



2

**CHEMICAL  
RESEARCH,  
DEVELOPMENT &  
ENGINEERING  
CENTER**

CRDEC-TR-318

**MID INFRARED POLARIZED LIGHT SCATTERING  
Applications for the Remote Detection  
of Chemical and Biological Contaminations**

**DTIC**  
**ELECTE**  
**S D D**  
MAR 13 1992

DETECTION DIRECTORATE

A. H. Carrieri  
D. J. Owens  
C. E. Henry  
K. E. Schmidt  
J. L. Jensen

J. R. Bottiger  
J. O. Jensen  
RESEARCH DIRECTORATE

C. M. Herzinger  
S. M. Haugland  
UNIVERSITY OF NEBRASKA AT LINCOLN  
Lincoln, NE 68508

'Original contains color  
plates: All DTIC reproduct-  
ions will be in black and  
white'

January 1992

Approved for public release; distribution is unlimited.

**U.S. ARMY  
ARMAMENT  
MUNITIONS  
CHEMICAL COMMAND**



Aberdeen Proving Ground, Maryland 21010-5423

92 000008

92-06523



**Disclaimer**

**The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorizing documents.**

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 1992 January	3. REPORT TYPE AND DATES COVERED Final, Dec 86 - Dec 91		
4. TITLE AND SUBTITLE Mid Infrared Polarized Light Scattering Applications for the Remote Detection of Chemical and Biological Contaminations		5. FUNDING NUMBERS PR-1C162706A553C		
6. AUTHOR(S) Carrieri, A.H., Owens, D.J., Henry, C.E., Schmidt, K.E.,* Jensen, J.L.,* Bottiger, J.R., and Jensen, J.O. (CRDEC), and Herzinger, C.M., and Haugland, S.M. (Univ. of Nebraska at Lincoln)				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  CDR, CRDEC, ATTN: SMCCR-DD/SMCCR-RSP-B, APG, MD 21010-5423 University of Nebraska at Lincoln, Lincoln, NE 68508		8. PERFORMING ORGANIZATION REPORT NUMBER CRDEC-TR-318		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  A polarized IR scattering facility that measures all elements of the Mueller matrix was built here at the U.S. Army Chemical Research, Development and Engineering Center (CRDEC) for remote detection of contaminants (analytes) spread on natural and manufactured surfaces (background scattering interferent). The ellipsometer is a two-modulator design that when interfaced to a mathematical algorithm can be trained to emit beam CO <sub>2</sub> laser energies at incident angles that best contrast various analyte target backscattering signatures from background signatures. For a probable detection event, sets of Mueller matrices are measured at beam energies coinciding with resonant IR absorptions by the analyte and at incident angle where backscattering by that analyte surface is strong. Identification of the contaminant(s) <u>in situ</u> is determined by another algorithm that operates on a vector whose $N \leq 16$ components are the independent Mueller elements measured at the susceptible beam energy/angle parameters. This work documents the phase sensitive detection program at CRDEC from past developments to future applications.				
14. SUBJECT TERMS Phase sensitive detection    Neural network Remote sensing                Vibrational Circular Dichroism Mueller matrix                (continued on page ii)			15. NUMBER OF PAGES 483	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

14. Subject Terms (continued)

Molecular absorption

Ellipsometer

Backscattering

## PREFACE

The work described in this report was authorized under Project No. 1C162706A553C, Reconnaissance, Detection, and Identification. This work was started in December 1986 and completed in December 1991.

The use of trade names or manufacturers names in this report does not constitute an official endorsement of any commercial products. This report may not be cited for purposes of advertisement.

Reproduction of this document in whole or in part is prohibited except with permission of the Commander, U.S. Army Chemical Research, Development and Engineering Center, ATTN: SMCCR-SPS-T, Aberdeen Proving Ground, MD 21010-5423. However, the Defense Technical Information Center and the National Technical Information Service are authorized to reproduce the document for U.S. Government purposes.

This report has been approved for release to the public.

## Acknowledgments

The authors recognize C. N. Merrow (STC, Incorporated, Hampton, VA) and D. Zeroka\* (Lehigh University, Bethlehem, PA) for suggesting the biological and controlled substance simulants compiled in Table 5. We also thank our colleagues E. Bahar (University of Nebraska at Lincoln, Lincoln NE), C. Chabalowski (U.S. Army Ballistic Research Laboratory), and H. Hameka (University of Pennsylvania, Philadelphia, PA) for stimulating and fruitful discussions on *full wave* scattering theory and on the *ab initio* calculations of quantum chemistry on molecular electronic structures, equilibrium geometries, and vibrational modes in the analyte materials.



Accession For	
NTIS	CRA&I <input checked="" type="checkbox"/>
DTIC	TAB <input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Code	
Dist	Availability Code
A-1	

\* Professor Zeroka is now involved in theoretical VCD predictions of sugar and amine compounds.

Blank

# CONTENTS

	page
1. INTRODUCTION .....	1
2. KEY DEFINITIONS USED IN PHOTOPOLARIMETRY .....	2
2.1 Stokes Vector .....	3
2.2 Mueller Matrix .....	4
3. EXPERIMENTAL APPROACH .....	9
3.1 Radiation Wavelength, Backscattering Angle, and Analyte Mass Density Dependencies of the Matrix Elements .....	10
3.1.1 Wavelength Selection of the Irradiating Probe Beams .....	10
3.1.2 Angle to the Sample of the Incident Probe Beams .....	10
3.1.3 Contaminant Deposition to the Sample .....	11
3.1.4 Time Dependence of Matrix Elements After Aerosol Ejection .....	13
4. THE ELLIPSOMETERS: THEORY OF OPERATION AND DESIGN .....	14
4.1 Hardware of the Experimental System .....	14
4.1.1 Probe CO <sub>2</sub> Lasers, Incident Beam Selection, Wavelength Monitor, and Power Regulation .....	17
4.1.2 Photoelastic Polarization Modulation .....	18
4.1.3 The Goniometer Beam Transceiver Arm For In Situ Analyses of Porous Wetted Surfaces .....	19
4.2 Ellipsometer System and Sub-System Matrices .....	21
4.3 MCT Detector Waveform of Scattered Radiation .....	22
4.4 Lock-in Detection Matrix of Primary and Overtone PEM Modulator Frequencies .....	29
4.5 Analog Amplitude and Phase Sensitive Detection Electronics .....	31
4.5.1 Mueller Matrix Acquisition: Theory of Operation .....	31
4.5.2 Reference Frequencies Generator, Mueller Elements Lock-in Amplifications .....	33
4.5.3 Digitization of the APSD Outputs (Mueller Matrix Channels) .....	33
4.5.4 Servo Loops for Variable MCT Gain Control and Incident Beam Power Regulation .....	34
4.5.5 Stepper Motor Control of Optics Hardware, Switching Between Incident Beams, PEM Peak Phase Retardation Selection Per Beam .....	36
4.6 Alignment and Calibration .....	36
4.6.1 Alignment of Coupled Polarization Modulator and Linear Polarizer Axes .....	36
4.6.2 Amplitude and Phase Adjustments of the Detector's Nine Element Channels to Transmission Optics of Known Mueller Matrix .....	38
4.6.3 Decoupling Sample From System Matrix Elements in the 3-Mirror Goniometer Type Ellipsometer Waveform Output .....	40

4.7	<i>System Software: Experiment Automation, Data Collection and Graphics Display</i>	46
4.7.1	<i>Switching of Laser Shutters, Modulator Retardation Adjustment</i>	46
4.7.2	<i>Modulator-Polarizer Permutations</i>	46
4.7.3	<i>Sample Selection and Stage Rotation</i>	46
4.7.4	<i>Goniometer Rotation and Data Acquisition</i>	46
4.7.5	<i>Data Storage and File Management</i>	46
4.7.6	<i>Graphics Presentation</i>	47
5.	<b>THEORETICAL MODELLING</b>	49
5.1	<i>Modelling the Analyte's Resonant Molecular Motions: Applying CADPAC, BROOKLYN89, and GAUSSIAN88 Quantum Chemistry Codes</i>	49
5.2	<i>Mueller Matrix Predictions by EM Wave Scattering From Rough Surfaces: The Full Wave Model of Physical Optics Theory and its Application for Remote Detection</i>	52
5.2.1	<i>The Full Wave Model: RETRO</i>	52
5.2.2	<i>DISPLAY: Graphical Analysis of the Full Wave Model</i>	55
5.2.3	<i>DETECT: Remote Detection Application of Full Wave Theory.</i>	73
5.2.4	<i>DECIDE2: A Detection Optimization Algorithm</i>	77
6.	<b>DISCUSSION OF FUTURE WORK</b>	78
6.1	<i>Digital Data Acquisition and Signal Processing of the Scattergram</i>	78
6.2	<i>Rapid Laser Switching Between Resonance-Reference and Resonance-Resonance Beam Wavelengths</i>	82
6.3	<i>Neural Network Computing of the Mueller Elements For Standoff Analyte Detections</i>	84
6.4	<i>Initial Experimental and Theoretical Mueller Matrix Data Bases</i>	88
6.4.1	<i>Metals and Insulators of Known IR Refractive Index and Surface Geometry: Validation of the Full Wave Model</i>	88
6.4.2	<i>Biological and Controlled Substance Simulants</i>	88
6.4.3	<i>Chemical Agent Simulants</i>	89
6.4.4	<i>Interferents</i>	89
7.	<b>CONCLUSION</b>	96
	<b>LITERATURE CITED</b>	97
	<b>APPENDIXES</b>	
	<b>I. EMISSIONS FROM CO<sub>2</sub> PROBE LASERS</b>	102
	<b>II. SYSTEM TO SAMPLE MUELLER MATRIX TRANSFORMATION: 3-MIRROR GONIOMETER TYPE ELLIPSOMETER FOR IN-SITU ANALYSES OF POROUS SURFACES</b>	114
	<b>III. APSD ELECTRONIC CIRCUITS: MODULES I-VIII</b>	121

IV.	<i>SOURCE CODE OF THE ANALOG APSD SOFTWARE MODULES .....</i>	<i>148</i>
V.	<i>RETRO/DISPLAY USER GUIDES AND SOURCE CODES .....</i>	<i>329</i>
VI.	<i>DETECT/DECIDE2 USER GUIDES AND SOURCE CODES .....</i>	<i>419</i>

## LIST OF FIGURES

1. Stokes Vector - Mueller Matrix Laser Beam Backscattering from a Particle Conglomerate .....	6
2. The Mueller Matrix and Some of its Typical Properties with Active Sub-field Elements .....	7
3. Spectra of the Imaginary Component of Complex Refractive Index (Absorption) in Volatile Liquid Chemical Agent Simulants Dimethyl Methyl (DMMP) and Diisopropyl Methyl (DIMP) Phosphonates, and Nonvolatile Polydimethylsiloxane (SF96 - General Electric Nomenclature) .....	12
4a-b. The Optical Arrangement of Ellipsometer Systems for Laboratory Study of (a) Dry/Wetted Surfaces of a Granular and Porous Texture, Like Terrestrial Surfaces, Undisturbed in a Natural Flat Position and (b) Dry/Wetted Surfaces of a Non-Porous Rigid Texture .....	15
4c. Field Evaluation of Contaminated Surfaces with a Prototype Ellipsometer Sensor .....	16
5. The Optical Orientations of Polarizer-Modulator Crystal Axis Producing the System Mueller Matrix Elements, as Measured from the Ellipsometer's Scattergram by the Analog Data Collection Unit .....	23
6. Flow Chart of the Analog Detection Electronics .....	32
7. The Ellipsometer's Optical System, Detector, and Gain-Control Modules .....	35
8. Alignment of the Ellipsometer's Linear Polarizer (POL1) - Photoelastic Modulator (PEM) Pairs .....	37
9. Calibration of the Analog Data Acquisition Channels for Measurement of all Mueller Matrix Elements .....	39
10. Calibration Experiments for Decoupling Four Mirror (M1-4) Matrix Elements from the Ellipsometer System Matrix of Figure 4a .....	42
11. A Right Angle Reflector (a) With a Single Mirror that Changes the Stokes Vector, and (b) with a Pair of Mirrors that do not .....	44
12. Graphics Output from the Software Package of Appendix IV .....	48
13. An Outline of the Computation Procedure Used to Predict the Analyte's IR Spectrum via Gaussian, CADPAC, and BROOKLYN Quantum Chemistry Software Packages .....	51
14a-f. Full-Wave Model Prediction of the Mueller Matrix Element $f_{11}$ in the Backscattering Direction from a Composite Soil Material Assuming a Surface Structure of Gaussian Distributed Slopes ( $\sigma$ ) and Heights (h) .....	57
15a-f. Full-Wave Model Prediction of the Mueller Matrix Element $f_{12}$ in the Backscattering Direction from a Composite Soil Material Assuming a Surface Structure of Gaussian Distributed Slopes ( $\sigma$ ) and Heights (h) .....	63

16. Regions of Irradiation Wavelength and Backscattering Angle in the $f_{11}$ Mueller Matrix Element that are Most Useful for Discriminating Against Liquid Chemical Agent Simulants SF96, DIMP, and DMMP on a Soil Surface .....	70
17. Regions of Irradiation Wavelength and Backscattering Angle in the $f_{12}$ Mueller Matrix Element that are Most Useful for Discriminating Against Liquid Chemical Agent Simulants SF96, DIMP, and DMMP on a Soil Surface .....	71
18. Regions of Irradiation Wavelength and Backscattering Angle in the $f_{34}$ Mueller Matrix Element that are Most Useful for Discriminating Against Liquid Chemical Agent Simulants SF96, DIMP, and DMMP on a Soil Surface .....	72
19. Mueller Matrix Digital Acquisition Using a Standard Analogic 6100/650 or LeCroy 9424 Waveform Analyser .....	80
20. Rapid 3-laser Switching Based on Variable Beam Splitter (VBS) Technology .....	83
21. A Neural Network Application for Accessing Elements of the Mueller Matrix in Real-Time Operation of the Ellipsometer Sensor .....	85
22. The Architecture of a Neural Network that Implements a Detection Algorithm for the Near Real-Time Identification of SF96, DMMP, and/or DIMP .....	86

## LIST OF TABLES

1. Normalized Stokes Vectors for Six Polarization States .....	4
2. The Mueller Matrices of Individual Optic Components Making up the 2-Modulator, Mid Infrared IR Ellipsometer Systems. ....	8
3. Lock-in Frequencies for Phase Sensitive Detection, and Corresponding Optical Alignments for Measurement of the Full Mueller Matrix. ....	30
4. Calibration of the PSD's Analog Electronic Channels to the Mueller Elements of a Rotating Quarter-Wave Plate $Q(\rho)$ , Rotating Linear Polarizer $P(\theta)$ , and Combination Optics $Q(\rho)P(\theta)$ and $P(\theta)Q(\rho)$ . ....	40
5. Candidates for Characterization Through Select IR Mueller Matrix Signatures.. ....	90

# MID INFRARED POLARIZED LIGHT SCATTERING

## Applications for the Remote Detection of Chemical and Biological Contaminations

### 1. INTRODUCTION

The work reported here originated with earlier measurements of bi-directional reflectance.<sup>1</sup> Those experiments measured the depolarized infrared reflectance component (a multiple-scattering effect referred to as volume reflectance) from contaminated soil, sand and other terrains and manufactured samples over the discrete energies of a grating-tunable CO<sub>2</sub> laser. We suspected the depolarized radiance to originate within the subsurface volume of the scattering substrate,<sup>2</sup> and be selectively absorbed by the interstitial liquid contaminant coatings when tuned through the analyte's IR resonance absorption frequencies. Consequently, the detected depolarized radiation component would have been transmitted through the liquid coatings and attenuated in exponential proportion to the product of coating thickness and absorption coefficient. Thus, the volume reflectance signals on resonance frequencies of the analyte can be related to film thickness of the contamination layers. (Separating the depolarized component reduces the problem to transmission of radiance from extended source, i.e. below the irradiation zone and through the contaminant layer.) We thus sought from the separated depolarized radiance a means to detect with finite probability (and certainly not uniquely) the presence of liquid contaminations on terrain, and approximate thickness of the coating layers through a set of differential volume reflectance measurements.

Our conclusion from analysis of these measurements was not encouraging. The minor depolarized IR scattering component (about 10% of the total scattering power) from agent simulant wetted soil and sand samples did not qualitatively reveal absorption by the contaminant at concentrations that would have proven fatal to life had it been an actual chemical agent. Furthermore, once absorption was detected at the highest concentrations, this method of detection could not singularly characterize the contaminant. It is unlikely that separate contaminants with different toxicity but overlapping extinction energies could be resolved through analysis of CO<sub>2</sub> laser reflectance spectra alone -- be it in total reflectance or its separated coherent and incoherent components.

New concepts were necessary, and in 1985 experimental and theoretical research programs were started on developing a phase-sensitive infrared scattering solution of chemical and biological warfare (CBW) agent detection problems. This new technology was certain to improve detection thresholds and provide quantitative information on physical and geometrical properties of the scatterer -- with interstitial contamination layers!! (We are developing techniques to use the rich phase information in the scattering EM waves that present reflectance-based systems, i.e., DISC/DIAL, do not or cannot measure.)

In the theoretical program, quantum chemistry codes are used to predict energies and absorption strengths of the contaminants at the molecular level. These infrared spectral intensities are converted into refractive indices that are accessed by a Full Wave electromagnetic wave scattering model used to predict the scatterer's 4x4 Mueller matrix. The Mueller matrix is a complete optical characterization of the scatterer, and is computed at beam excitation energies and backscattering angles that can best contrast a contaminant (referred to as the chemical analyte with an IR absorption moiety) from all other scatterers (background). In the experimental program, we are developing three ellipsometer sensors for production of a data base of Mueller elements representing scattering by aerosols and by liquid coatings spread across various surfaces. With guidance from a valid theoretical scattering model, the ellipsometer sensors can be made to operate at beam energies and backscatter angles that produce sets of Mueller elements that are susceptible to the contaminant and only that contaminant. The set of independent elements most sensitive to the analyte(s) are inputs to an algorithm designed to identify it, or establish non-presence.

The purpose, then, of this work is to characterize contaminated surfaces *in toto* through their Mueller matrix signatures, interpret these data elements, discern targeted contaminant information immediately (near real-time alarm), and quantify the threat target mass concentration (map). The analyte compounds of interest include chiral sugars and other enantiomers that preferentially absorb right- and left-circularly polarized light, and thus simulate Vibrational Circular Dichroism (VCD) in more complex biological structures. Background (interferent) materials of interest include manufactured and terrestrial (scattering) interferent surfaces such as soil, sand, concrete, asphalt, and treated metallics commonly used in military hardware. Other analytes we wish to target for identification are the chemical agent simulant class of phosphonated hydrocarbons and other liquids that exhibit at least one strong resonant IR vibrational normal mode.

Presented in Section 2 are fundamental definitions of photopolarimetry on which the ellipsometer is based. In Section 3, the types of measurements to be conducted and the important experimental parameters are discussed. In Section 4, the ellipsometer and its theory of operation is presented, and in Section 5, a Full-Wave light scattering model for rough surface scattering is introduced, currently under development at the University of Nebraska in collaboration with CRDEC. In this same section, three quantum chemistry software packages are briefly reviewed. These *ab initio* models predict least-energy group configurations in the analyte molecules, and their corresponding absorption spectrum. We will merge quantum molecular codes with the Full Wave polarized EM wave scattering code, so that a comprehensive model can be used to predict linear and circular birefringence; VCD, depolarization, and other polarization dependent scattering phenomena.

The integrated Full Wave scattering and quantum molecular codes will be tested through a systematic set of experiments and if found valid guide the development of a field ellipsometer sensor capable of multi-target detections, by simulating the entire experiment under various field scenarios. These simulations of Mueller backscattering elements will direct us toward optimizing those parameters most crucial in development of a prototype version of the 9-channel analog laboratory phase sensitive detection system (first generation, Section 4), a digital data acquisition system counterpart (second generation, Section 6.2), and neural network (third generation, Section 6.4). In Section 6 and the concluding Section 7, present development status of the experimental ellipsometer systems is updated, methods of advanced digital data acquisition and processing techniques are suggested for a future detection module of the ellipsometer, the structure of an initial data base is outlined, and a brief discussion on our initial work with neural network computing is addressed.

## 2. KEY DEFINITIONS USED IN PHOTOPOLARIMETRY

References 3 through 7 cover in some detail the definitions and conventions used in ellipsometry. Among these, Shurcliff's book<sup>4</sup> describes polarized light best at the introductory level, and includes useful Mueller matrices for standard birefringent and polarizing optics that make up these ellipsometer instruments. The standard texts by van de Hulst<sup>5</sup>, Bohren and Huffman<sup>9</sup> are frequently referenced sources for Mie and Rayleigh scattering by arbitrary particles. Moreover, Section 5 is reserved for a brief but more focused theoretical treatment of the Mueller elements related to scattering by dielectric and metal surfaces of varied roughness (changing heights and slopes), a subject we later concentrate on for this detection problem.

We begin our discussion by defining a Stokes vector and Mueller matrix: a column vector representing total and partially-polarized states of propagating electromagnetic laser radiation, and a matrix that transforms these states due to reflection, transmission, scattering, or absorption events. They represent the fundamental operational principles of the ellipsometer instruments. Notation of J.D. Jackson<sup>10</sup> in defining the Stokes parameter is used throughout this work. We also choose  $f$  symbols for Mueller matrix elements of the scattering sample, as denoted by R.C. Thompson, J.R. Bottiger and E.S. Fry.<sup>11</sup>

## 2.1 Stokes Vector.

The Stokes vector is a four-element column matrix that represents a beam of total or partially-polarized light. Consider a homogeneous plane wave propagating in a direction of the vector  $\mathbf{k} = n\hat{\mathbf{k}}$ ,

$$\mathbf{E} = (E_1\hat{\mathbf{e}}_1 + E_2\hat{\mathbf{e}}_2)e^{i(\mathbf{k}\cdot\mathbf{r}-\omega t)} \quad (1)$$

where the complex electric field amplitudes,  $E_1$  and  $E_2$ , have amplitude ( $a$ ) and phase ( $\delta$ ).

$$E_1 = a_1 e^{i\delta_1}, \quad E_2 = a_2 e^{i\delta_2} \quad (2)$$

Elements of the Stokes column matrix are defined in terms of the coordinate basis vectors  $\hat{\mathbf{e}}_1$  and  $\hat{\mathbf{e}}_2$  as:

$$\mathbf{s} = \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} \langle |\hat{\mathbf{e}}_1 \cdot \mathbf{E}|^2 \rangle + \langle |\hat{\mathbf{e}}_2 \cdot \mathbf{E}|^2 \rangle \\ \langle |\hat{\mathbf{e}}_1 \cdot \mathbf{E}|^2 \rangle - \langle |\hat{\mathbf{e}}_2 \cdot \mathbf{E}|^2 \rangle \\ \langle 2\text{Re}(\hat{\mathbf{e}}_1 \cdot \mathbf{E})(\hat{\mathbf{e}}_2 \cdot \mathbf{E}) \rangle \\ \langle 2\text{Im}(\hat{\mathbf{e}}_1 \cdot \mathbf{E})(\hat{\mathbf{e}}_2 \cdot \mathbf{E}) \rangle \end{pmatrix} = \begin{pmatrix} \langle a_1^2 \rangle + \langle a_2^2 \rangle \\ \langle a_1^2 \rangle - \langle a_2^2 \rangle \\ \langle 2a_1 a_2 \cos(\delta_2 - \delta_1) \rangle \\ \langle 2a_1 a_2 \sin(\delta_2 - \delta_1) \rangle \end{pmatrix} \quad (3)$$

\* A linearly-polarized beam that is multiply scattered from terrestrial surfaces will have a depolarized component in the IR.<sup>1</sup> The Stokes calculus is necessary when predicting these transformations.

Brackets  $\langle \rangle$  denote time-averaged values. Notice that  $s_0$  is an additive intensity term and suggests total scattering power, while positive (negative)  $s_1$  suggests a majority of horizontal (vertical) linear polarization. Both  $s_2$  and  $s_3$  contain phase difference terms, and can suggest + (clockwise) or - (counter clockwise) handedness of elliptical EM waves. As an illustration, consider the Stokes parameters for three polarization states: (a) *unpolarized*<sup>\*\*</sup>, (b) horizontal linear and (c) right circular. For case (a), by definition of unpolarized light,  $\langle |\hat{e}_1 \cdot E|^2 \rangle = \langle |\hat{e}_2 \cdot E|^2 \rangle$ , and  $s_3 = s_4 = 0$ , since both sine and cosine terms average to zero independently of the amplitudes  $a_1$  and  $a_2$ . When dividing all parameters by  $s_0$ , the normalized Stokes vector for unpolarized light becomes (1,0,0,0). For case (b),  $\delta_1 = \delta_2$ ,  $a_2 = 0$ ,  $\hat{e}_2 \cdot E = 0$ , and therefore the normalized Stokes vector for linearly polarized light in the horizontal direction is (1,1,0,0). For the final case (c),  $\delta_1 = \delta_2 - \frac{\pi}{2}$ ,  $a_1 = a_2$ , and therefore the normalized Stokes vector for right circular polarized light is (1,0,0,1).

The Stokes vector of the incident beam and the scattered radiance, collected from the irradiation zone in some small solid angle from the sample, changes continuously and periodically when operated on by the ellipsometer's transmitter and receiver photoelastic modulation (PEM) optics, respectively. The following Table 1 lists a selection of six states generated by specific retardation amplitudes in the PEMs, and swept periodically at 33.980 KHz (transmitter) and 31.896 KHz (receiver) transducer frequencies, rates at which the PEM's octagonal ZnSe crystal are stressed-then-relaxed. We later make reference to this table when the topics of system matrices and diagnostics of MCT detector signal are discussed.

Table 1. Normalized Stokes vectors for six polarization states.

HORIZONTAL LINEAR	VERTICAL LINEAR	+45° LINEAR	-45° LINEAR	RIGHT CIRCULAR	LEFT CIRCULAR
$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}$

## 2.2 Mueller Matrix.

The polarization state of an electromagnetic wave is generally altered by reflection, transmission, scattering and/or absorption processes. The scattering phase and amplitude information represented by the Mueller matrix depends on the physical properties of the interactive medium and its geometry or topographic detail. A transformation of the incident beam Stokes vector after its backscattered interaction with a material boundary into a new vector defines the Mueller system matrix operator.

<sup>\*\*</sup> Unpolarized light implies nonpreferential electric field directional properties, i.e., the total electric field vector is equally probable of lying in any orientation in the scattering plane over the time in which a measurement is made.

### The Mueller Matrix Transforms the Input Stokes Polarization State Vector.

$$s_j = f_{jk} s_k \quad (4)$$

In Equation (4),  $s_k$  are Stokes vector components of the incident beam, and  $s_j$  are resultant components after beam - medium interaction. That transformation is given by  $f_{jk}$ , a 4x4 operator whose elements represent a complete geometrical and physical description of a linear medium interacted by the beam - it is the Mueller matrix. (The non-linear phenomena of stimulated Raman and Brillouin scattering, second harmonic generation, etc., cannot be interpreted by a linear operator. The medium may require a tensor description of permittivity and (for magnetic materials) permeability. However, the electric field intensity required to produce such effects is far beyond the incident irradiations by these ellipsometer probe beams.) In Figure 1, we graphically illustrate the beam scattering geometry. The Mueller matrix field and a few of its elemental interpretations are schematized in Figure 2.

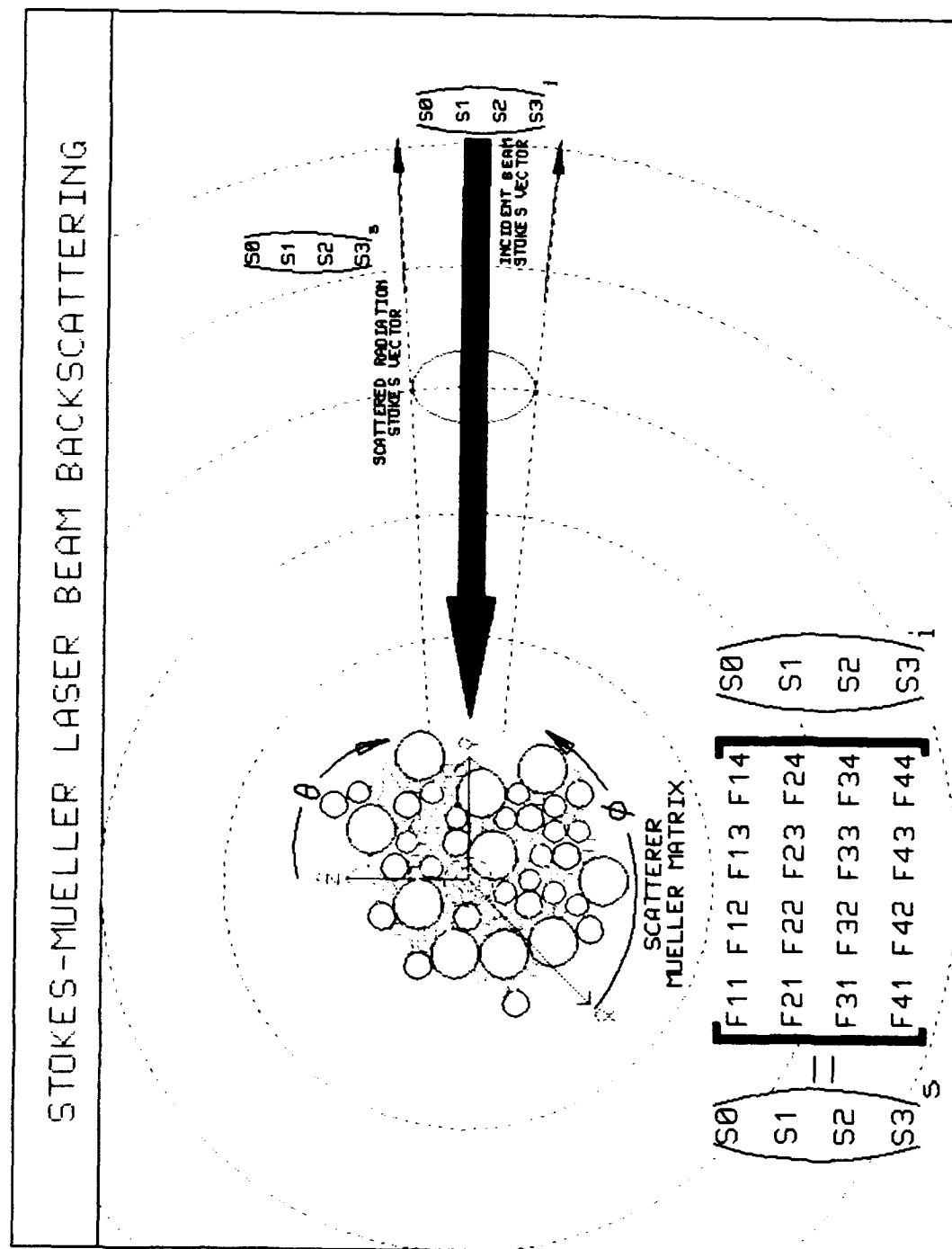


Figure 1. Stokes vector - Mueller matrix laser beam backscattering from a particle conglomerate.

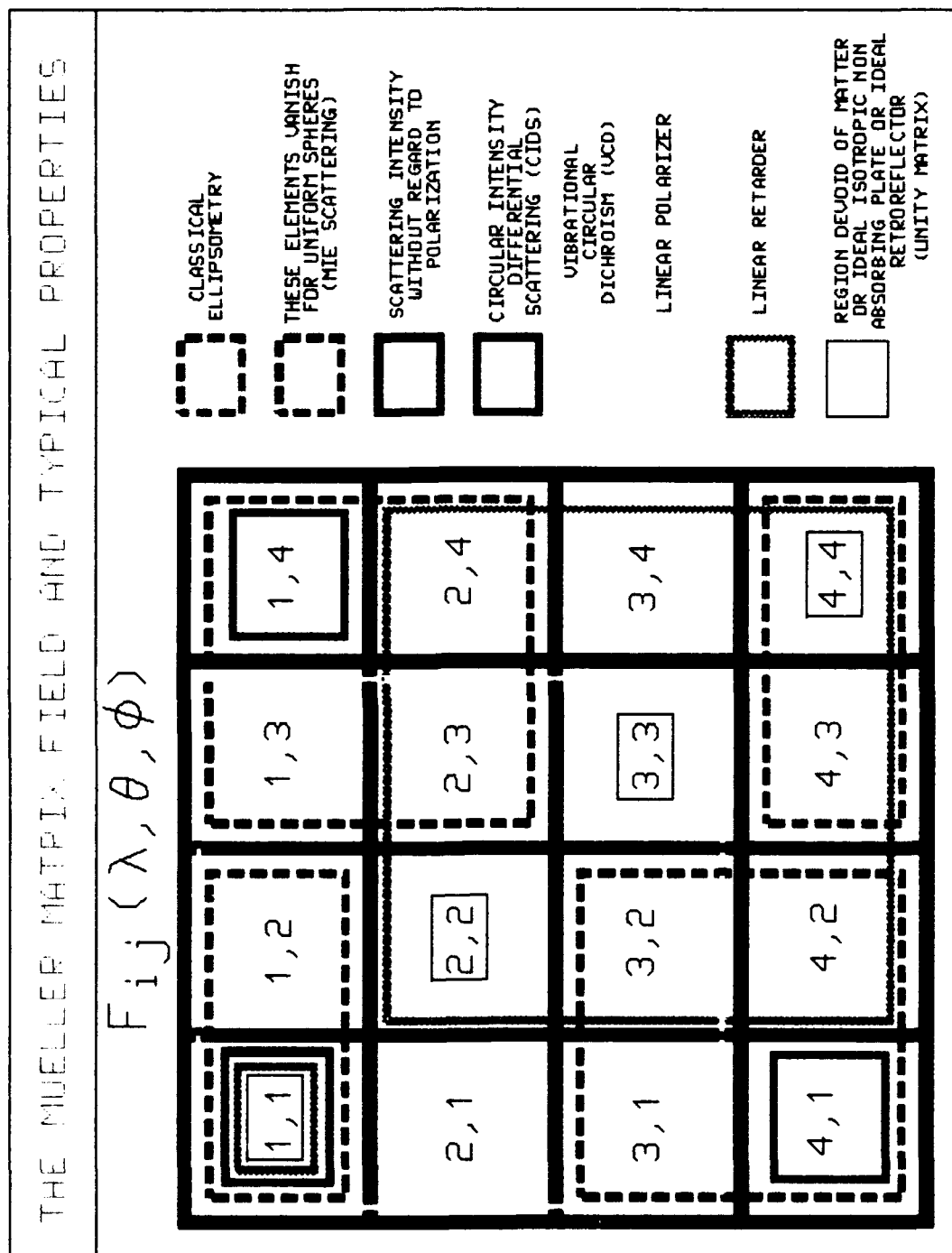


Figure 2. The Mueller matrix and some of its typical properties with active sub-field elements.

In the following Table 2, Mueller matrices of the scattering sample (denoted F, the surface and subsurface contaminant layers are embedded in F) and optical components of the ellipsometer design are presented. The matrix elements of reflection, retardation, and polarization optics are used in latter sections of this work when all Mueller elements of the contaminated sample are correlated to primary and combination PEM modulator frequencies in the instrument's Fourier transformed MCT detector signal intensity spectrum.

Table 2. The Mueller matrices of individual optic components making up the 2-modulator, mid infrared IR ellipsometer systems.  $f_{ij}$  are matrix elements of the scattering sample,  $\alpha$  and  $\beta$  are amplitude ratios of reflected-to-incident electric field components for light polarized parallel and perpendicular to the plane of incidence, respectively,  $\sigma$  is the reflection induced phase shift between these two components,  $\theta$  is the polar (tilt) angle resulting from goniometer rotation,  $\delta_0$  and  $\omega$  are peak retardance and frequency generated by the ZnSe phase modulators, respectively.

MUELLER MATRICES OF THE EXPERIMENTAL SYSTEM			
Scattering Sample			
$\begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ f_{41} & f_{42} & f_{43} & f_{44} \end{pmatrix}$			
Goniometer			
Mirror		Rotation operator	
$\frac{1}{2} \begin{pmatrix} \alpha^2 + \beta^2 & \alpha^2 - \beta^2 & 0 & 0 \\ \alpha^2 - \beta^2 & \alpha^2 + \beta^2 & 0 & 0 \\ 0 & 0 & -2\alpha\beta\cos\sigma & 2\alpha\beta\sin\sigma \\ 0 & 0 & 2\alpha\beta\sin\sigma & -2\alpha\beta\cos\sigma \end{pmatrix}$		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 2\theta & \sin 2\theta & 0 \\ 0 & -\sin 2\theta & \cos 2\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	
Linear Polarizers			
Horizontal(Vertical)		+(-)45 Degrees	
$\frac{1}{2} \begin{pmatrix} 1 & +(-)1 & 0 & 0 \\ +(-)1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$		$\frac{1}{2} \begin{pmatrix} 1 & 0 & +(-)1 & 0 \\ 0 & 0 & 0 & 0 \\ +(-)1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	
Photoelastic Modulators			
Vertical(Horizontal)		+(-)45 Degrees	
$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \cos(\delta_0\cos\omega t) & -(+)\sin(\delta_0\cos\omega t) \\ 0 & 0 & +(-)\sin(\delta_0\cos\omega t) & \cos(\delta_0\cos\omega t) \end{pmatrix}$		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\delta_0\cos\omega t) & 0 & +(-)\sin(\delta_0\cos\omega t) \\ 0 & 0 & 1 & 0 \\ 0 & -(+)\sin(\delta_0\cos\omega t) & 0 & \cos(\delta_0\cos\omega t) \end{pmatrix}$	

### 3. EXPERIMENTAL APPROACH

In addition to Reference 11, the works of Williams<sup>12</sup>, Roseler<sup>13</sup> and Vorburger et al.<sup>14</sup> seem relevant to this specific detection problem. Williams describes depolarization, cross polarization, and changes in ellipticity through rough surface scattering signatures by a Poincare sphere representation, while Roseler introduces a Fourier transform spectrometer formalism to yield a spectroscopic IR phase matrix measurement. Vorburger et al., discuss the ellipsometric parameters  $\psi$  and  $\Delta$  for various textured surfaces, presenting data at 0.6328  $\mu\text{m}$  and 0.5461  $\mu\text{m}$ . They conclude that topographical roughness is a major random error source for inversion methods that map  $\psi$  and  $\Delta$  parameters to physical properties of the scatterer. This is a major concern if polarized scattering can be successfully applied to the IR remote sensing problem: for a quantitative in situ infrared detection of amorphous or crystalline contaminants, its physical absorption property (where the imaginary part of its permittivity maximizes) is its key identifier, and that information must manifest itself consistently in the susceptible on/off resonance differential scattering Mueller elements. This issue is addressed again in Section 5.

From our most recent literature search, we have located other visible single-frequency bistatic photopolarimeter systems of one and four modulator design<sup>15</sup>, with essentially the same detector analog electronics as first reported by Kemp<sup>16</sup>, and later Hunt and Huffman.<sup>17</sup> The work of Whitt and Ulaby's<sup>18</sup> is also noted, reporting millimeter-wave polarimeter measurements, as is van Zyl's<sup>19</sup> paper on radar polarization signatures from rough surfaces in the backscattering plane.

We have not located in the open literature data published on mid IR Mueller matrix elements from rough surfaces of varied optical properties, e.g., the contaminated terrestrial scenarios. The instruments we report here were designed especially for this application, and provide an opportunity to acquire quantitative Mueller data elements, sensitive to perhaps sub-micron layers of contaminant spread across a surface, and conduct the necessary near real-time data processing and analysis that we hope will rapidly identify (alarm) and quantify (map) chemical and biological warfare (CBW) agents dispersed onto terrestrial and manufactured materials.

The ellipsometer remote sensors presented here consist of two photoelastic modulators driven at frequencies offset by 2.06 KHz, one operating on the incident beam the other on collected backscattered radiance. They are the Mueller matrix generators capable of optically computing nine elements simultaneously and all sixteen in sequence according to orientations of modulator-polarizer axes. One instrument will be used to analyze surfaces of a rigid and generally continuous texture; a nonporous military painted panel for instance. A similar laboratory instrument will analyze the natural soils and other granular porous surfaces without mechanical disturbance to the scattering sample. This instrument requires a goniometer optical device. A third instrument is intended for long-range field evaluation.

In conducting laboratory Mueller matrix measurements from soil and other granular-like samples, in situ, an opto-mechanical goniometer arm was built and later integrated into one ellipsometer optical design, between linear polarizing optics producing the system's initial and final Stokes vectors. The goniometer introduces three mirror optics to the ellipsometer's system matrix, a necessary but optically and computationally complex addition to the ellipsometer system and its data analysis functions. As we progress toward later sections of this work, the data analysis complications that arise as a result of the presence of these mirror optics will be made more clear. Moreover, we later present both software and hardware methods for compensating all goniometer mirror phase contributions so that the desired sample matrix elements is extracted from the detector system waveforms.

The order in which the incident beam is transmitted and reflected by system optic components, and scattered by the sample, defines the important system product matrix. Only in the goniometer-type and field instruments do system (measured) elements need to be transformed into sample elements ( $f_{ij}[\lambda, \theta]$ ) by software and electronic decoding ( $f'_{ij}[\lambda, \theta]$ , Appendix II). These sample Mueller elements contain the important information on the analyte that we seek in making detection judgements.

### 3.1 Radiation Wavelength, Backscattering Angle, and Analyte Mass Density Dependencies of the Matrix Elements.

The experimental goal is production of a reliable data bank of Mueller matrix elements that most contrast background (terrain) and chemical/biological contaminant (analyte). These elements will be obtained as functions of wavelength of the probe beams; tuned specifically to the analyte, angle in the backscattering plane of the incident beams, and mass of contaminant deposited to the sample per unit area in the zone of beam exposure.

We reserve the latter Section 6.4 for a discussion of our starting data base structure and its management.

#### 3.1.1 Wavelength Selection of the Irradiating Probe Beams.

The Mueller matrix elements are to be measured alternately at CO<sub>2</sub> beam energies that drive strong vibrational modes in the contaminant to be detected (sometimes referred to as analytical wavelengths) and at an energy where no contaminant molecular excitation is produced (reference).

In addition to a standard C<sup>12</sup>O<sub>2</sub><sup>16</sup> laser source, isotopic carbon dioxide gases are part of the gain fill in three other lasers of the ellipsometer transmitter. The isotope lasers are used for purposes of extending the mid IR wavelength range in which the sample can be irradiated, and to fill in wavelength gaps between P- and R-branch C<sup>12</sup>O<sub>2</sub><sup>16</sup> transitions (widen and make more continuous the wavelength coverage). The beam wavelength selections (i.e., emissions with enough power for measuring scattered radiance) range from 9.0  $\mu\text{m}$  at the R(40) line 00<sup>0</sup>1-02<sup>0</sup>0 band in the C<sup>12</sup>O<sub>2</sub><sup>16</sup> laser, to 12.08  $\mu\text{m}$  at the P(44) line 00<sup>0</sup>-10<sup>0</sup>0 band of the C<sup>14</sup>O<sub>2</sub><sup>16</sup> laser. For example, consider detection of the liquid chemical agent simulant DIMP, CH<sub>3</sub>PO(OCH(CH<sub>3</sub>)<sub>2</sub>)<sub>2</sub>. Absorption band assignments of this analyte are two intense  $\nu$  (P-O-C) vibrational modes at 10.169  $\mu\text{m}$  and 9.884  $\mu\text{m}$ , and a less intense P-CH<sub>3</sub> rocking mode at 10.902  $\mu\text{m}$ . Typically, three of four lasers will be tuned to wavelengths that align to peak maximum in the vibrational bands (analytical wavelengths) of the target, while the fourth laser is off-tuned to generate reference (background) Mueller data elements (Appendix I). The full Mueller matrix 16-element field recorded between beams and those that possess susceptible behavior to the analyte (change abruptly during laser switching between resonance and reference wavelengths) are singled out as detection candidates.

#### 3.1.2 Angle to the Sample of the Incident Probe Beams.

Generally, standoff active sensors are monostatic backscattering systems where the transmitter source and collection optical receiver are stationary and co-located as a unit. The ellipsometer facility at CRDEC is also monostatic, measuring the Mueller elements in backscattering directions over all angles of incidence. To accommodate such measurements from granular materials and surfaces of continuous texture, two separate ellipsometers were constructed and are schematized in Figures 4a-b. For the rigid surfaces, like painted metallics and

pelletized substrates, the sample is positioned vertically and rotated along an axis perpendicular to the plane of incidence defined by the incident beam at polar angles ranging from  $-89^{\circ}$  to  $+89^{\circ}$  (Figure 4b). For loose particles, like natural soil and sand, the sample lies undisturbed in its horizontal position while a goniometer transceiver arm delivers the beam to the surface over all polar angles in the upper hemisphere and directs backscattering to the MCT detector (Figure 4a) by use of three flat mirrors, each positioned  $45^{\circ}$  to its incident beam. Thus, the reference plane in which the Stokes vectors of transceived radiations are measured vary as the goniometer arm rotates out of the reference plane – a complication this design presents in data analysis. Introduction of these mirrors between linear polarizers producing incident- and final-Stokes vectors of the ellipsometer design causes other complications. Matrix signatures from the sample must now be separated from those elements by the ellipsometer system. The phase changes imparted by each goniometer mirror optic can be compensated for either optically, or mathematically through a series of calibration experiments (Section 4.6.3, and Appendix II).

A phenomenon of direct backscattering from optically rough surfaces is the so-called **opposition effect**, which may have benefit for these backscattering ellipsometers as detection instruments. The effect is predicted by Full Wave theory (see Section 6), and states that an enhanced incoherent *backscattering* component of radiance results from scattering at a randomly rough boundary interface. An experiment conducted by Mendez and O'Donnell<sup>20</sup> show results that tend to confirm this prediction. Bohren and Huffman<sup>9</sup> show that if the irradiated surface is modeled as a random array of identical dipole oscillators, then the total backscattering radiation field is generally incoherent and dependent on spacing between the dipole radiators, unlike the forward scattering field that is totally in phase and independent of separation between radiating particles. Since enhanced backscattering manifests itself as a noncoherent component of scattering, it would appear, then, that a spatially integrated backscattering signal from randomly rough surfaces could vary dramatically in the scattering phase signature (*viz*, fluctuations in the Mueller matrix backscattering 'picture' as the select irradiation beam spatially scan a contaminated terrestrial surface.) Again, we emphasize that a changing refractive index in the analyte between beams alternating on target resonance and off target absorption resonance must be revealed in a reproducible manner for making a successful, unique, physical detection.

### 3.1.3 Contaminant Deposition to the Sample.

Three common liquid simulants are chosen as initial detection targets: nonvolatile dimethyl siloxane fluid SF96 (General Electric nomenclature), and the more volatile DMMP and DIMP phosphonated organic compounds. Presented in Figure 3 are infrared imaginary refractive indices of all three compounds, and listed in Table 4 are the compounds' strongest mid IR extinction frequencies. (The imaginary part of refractive index is proportional to the absorption coefficient of the medium. It can be obtained through Kramers-Kronig analysis of either absorption or reflectance spectra<sup>10,21</sup>.)

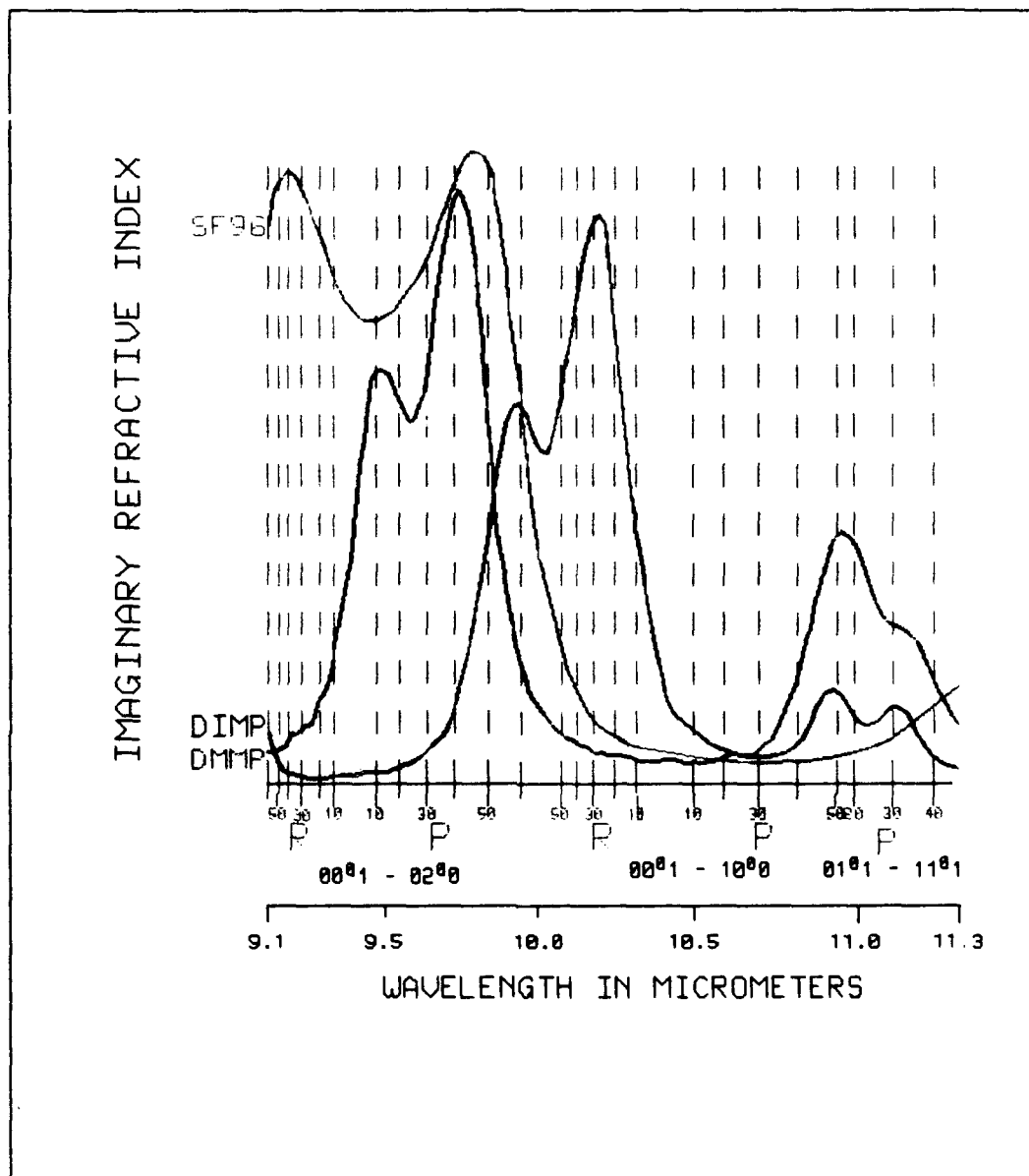


Figure 3. Spectra of the imaginary component of complex refractive index (absorption) in volatile liquid chemical agent simulants dimethyl methyl (DMMP) and diisopropyl methyl (DIMP) phosphonates, and nonvolatile polydimethylsiloxane (SF96 - General Electric nomenclature). The dashed vertical lines are some of the allowed CO<sub>2</sub> laser transitions where the ellipsometer can probe. See Appendix I for a complete list of transition assignments and nomenclature for these ellipsometer's beam sources.

The relationship of applied mass contamination on a surface and the Mueller elements, at angle-wavelength domains specific for detection of the contaminant, could yield useful information to a trained signal processor for quantification of the analyte once detection is established from an appropriate data bank.

This data bank of information can come from a set of experiments that sequential measure all Mueller elements from some scattering substrate when dry, then coated with increasing quantities of the analyte. Consider a soil substrate as the background scatterer. First, all sixteen matrix elements are measured from the unwetted soil sample and referenced as the background matrix set of Mueller elements over a range of beam energies. Next, the soil is contaminated by the analyte compound in low concentration and the elements are once again measured, at beam energies specific to the analyte. The analyte-susceptible Mueller elements are singled out for discriminant features, then the sample is re-wetted with additional analyte solution and the matrix elements remeasured. Measurements continue to an analyte density typically less than  $20 \text{ gm cm}^{-2}$ . These kinds of controlled analyte dispersion experiments will correlate a pattern in the analyte-susceptible Mueller elements from low to high analyte mass density, with the uncontaminated elements serving as a reference frame of background information. The element data can subsequently be input to an appropriate decision making algorithm (Appendix VI) that electronically filters background data and discerns the informational content of the analyte-specific Mueller element signals, correlating their pattern to quantity of analyte spread across/into the surface. In a typical data trial, contaminant mass densities of  $2\text{-}20 \text{ gm m}^{-2}$  are deposited in  $2\text{-}5 \text{ gm m}^{-2}$  increments.

Experiments for classifying biological contaminants spread across terrain and manufactured surfaces through their Mueller matrix features are being discussed now. We feel some control experiments must first be performed to first recognize whether resonant absorption by these analytes can reveal a *matrix* signature in the absence of interferent scatterers. Initial experiments would include measurements from aqueous suspensions of *simulant* organisms, generated bioaerosols in a chamber, and liquid and crystalline compounds like sugars that have known molecular symmetrizations exhibiting dichroism (e.g., vibrational circular dichroism or VCD). If features of detection are clearly evident in differential Mueller element data sets then the experiments are refined (i.e., concentrate on the optimum angle-wavelength domains producing susceptible Mueller elements), redone with interferent scattering (an *in situ* contaminated surface scenario), then analyzed for the element features separate from the terrain element signatures. The experiments would be extended to include other disseminated specimens like sterile *B. Anthracis*, *B. Cereus*, *B. Thuringiensis*, *E. Coli*, and fungus spores.

#### 3.1.4 Time Dependence of Matrix Elements After Aerosol Ejection.

In the experimental program we monitor the susceptible Mueller elements after deposition of the contaminant. The expected temporal fluctuations in these Mueller elements could conceivably sense diffusion and spread of contaminant across and into the substrate. Evaporation of the volatile liquid DMMP and DIMP analytes would also manifest temporal fluctuations in these elements. Heat liberated by an absorbed beam energy will deplete the thin volatile analyte surface coatings, thereby generating a contaminant vapor cloud above the irradiation zone. It is difficult to predict whether this vapor presence above the surface will cause significant alterations in the ac (phase spectrum) scattering components. (Backscattering from the vapor itself is insignificant, however, scattering from the terrain transmitting through the vapor cloud may indeed be sensitive to some matrix elements.) Transmission of scattered radiance through the vapor plume on analyte resonant wavelengths would have, however, some attenuation effect on  $f_{11}$ , i.e., the dc component of the MCT signal, it being a measure of scattering intensity.

Other changes in  $f_{ij}$  from volatile simulants, if indeed they are measurable, are likely to result from changing scattering surface topography due to evaporation and diffusion of liquid analyte layers or, in biological specimens, a changing morphology under exposure by an absorbing beam.

#### 4. THE ELLIPSOMETERS: THEORY OF OPERATION AND DESIGN

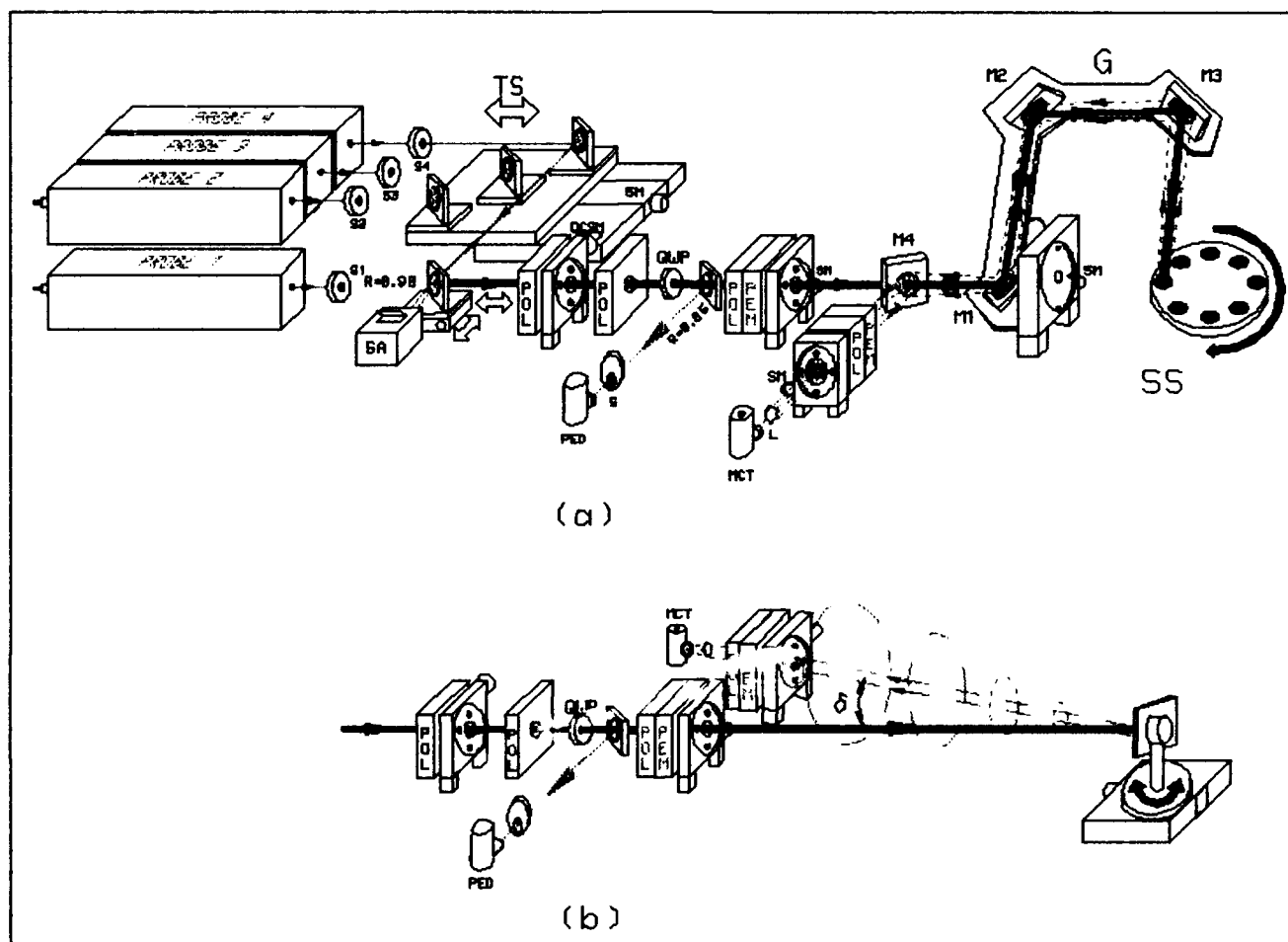
The CRDEC Mueller matrix ellipsometer (MME) systems are similar in optic and analog data acquisition system designs to the facility reported in Reference 11. Our experimental systems are multi-laser and multi-infrared wavelength (9.0-12.1  $\mu\text{m}$ ), monostatic backscattering, and of a two modulator design; theirs a single-visible wavelength (0.6328  $\mu\text{m}$ ), bistatic, four modulator system. A disadvantage of these two-modulator systems is that only 9 of 16 matrix elements can be simultaneously measured. It requires four sequential angular permutations between each of this ellipsometer's linear polarizer-retarder optic units (Section 4.4) to complete the measurement of the 16-element field. We found it necessary, for precision and repeatable measurements, to stepper-motor control and computer automate all optic translation and rotation sequences. It follows, that computer data collection operations are synchronous to indexing of the optics hardware.

Advantages of the two- versus four-modulator systems include less multiplicative error in the modulation crystals due to: (a) imperfections in the crystal, (b) thermal distortion by heating by the incident IR beam, (c) misalignments in optics, and (c) beam transmission offset from the PEM crystalline axis. A less chance of mixing of the Mueller elements (i.e., overlap of the Fourier intensities in the transformed scattergram) is realized because of the wider separation between primary and overtone modulator frequencies. (The requirements imposed on the phase-sensitive detector boards are less restrict because of the greater frequency separation, and thus less chance of harmonic overlap, between Fourier signal components.)

##### 4.1 Hardware of the Experimental System.

Figures 4a-c shows the ellipsometers' basic hardware components for three types of experiments: (a) short range laboratory matrix element measurements from dry/wetted porous granular materials (the soils and sands) left undisturbed; (b) short range laboratory matrix measurements from surfaces of nonporous composition made to rotate over all backscattering angles (the flat, continuous, and cohesive surfaces like painted metallic panels used in military hardware); and (c) long range matrix measurements, where the beam is sent outside the laboratory to target boards located down range at distances of 500 meters and more. (The switching system of this system's transmitter is reviewed in Section 6.2.) All systems are automated and computer controlled, including: laser switching, beam and sample positioning, power regulation, goniometer and PEM-polarizer rotation; data acquisition, storage, and processing. Wavelength selection and stabilization of the four laser systems is part of an initialization procedure, the only manual optics hands-on task required by the operator of these instruments. (This may also be automated in future prototype systems if tunable laser sources are required.)

The salient components of the ellipsometer system are now discussed.



Figures 4a-b. The optical arrangement of ellipsometer systems for laboratory study of (a) dry/wetted surfaces of a granular and porous texture, like terrestrial surfaces, undisturbed in a natural flat position and (b) dry/wetted surfaces of a non-porous rigid texture. The components of the ellipsometer instrument include: Probes 1-4, four mid infrared lasers with distinct  $\text{CO}_2$  gain media (three are isotopic); S1-S4, shutters intercepting the four incident beams; TS translation stage to direct the appropriate beam to the scattering sample; SM, stepper motors providing accurate computer-controlled stage rotations and translations; SA, spectrum analyser for determining beam wavelength; DCSM, dc servo motor for power regulation of the incident beams; POL, linear polarizers; QWP, quarter-wave plate for production of circularly-polarized radiation; PEM, photoelastic modulators; M1-M4, mirrors for reflecting incident radiation  $45^\circ$ ; G, goniometer beam transceiver arm; SS, rotary sample stage; L, focusing lens to the MCT detector chip; PED, pyroelectric detector for monitoring the incident beams;  $\delta$ , a small deviation from the backscattering angle; and MCT, mercury-cadmium-telluride detector of scattered radiations.



#### 4.1.1 Probe CO<sub>2</sub> Lasers, Incident Beam Selection, Wavelength Monitor, and Power Regulation.

Four sealed CO<sub>2</sub> continuous-wave (cw) lasers make up the transmitter section of the ellipsometers. Each laser is grating wavelength tunable, and contains piezoelectric circuitry and feedback for mode-locking and beam amplitude and frequency stabilities. The gain media of the four lasers are inert gases mixed with, respectively, nonisotopic C<sup>12</sup>O<sub>2</sub><sup>16</sup> and isotopic C<sup>13</sup>O<sub>2</sub><sup>16</sup>, C<sup>14</sup>O<sub>2</sub><sup>16</sup>, and C<sup>12</sup>O<sub>2</sub><sup>18</sup> gases. Advantages of this four-laser configuration include: (a) a wider wavelength range to probe the sample; (b) a greater selection of discrete wavelengths to probe the surface, therefore, more continuous spectral coverage in the mid IR region; and (c) rapid 4-wavelength matrix measurements without a need for retuning and res- tabilization. Both (a) and (b), above, follow because vibrational-rotational excited states of the more massive triatomic CO<sub>2</sub> isotopes have shifted P- and R-branch transitions beyond broadening of the C<sup>12</sup>O<sub>2</sub><sup>16</sup> laser lines.

For the reader interested in CO<sub>2</sub> spectroscopy, we refer to Herzberg's<sup>22</sup> standard text. For a description of the principal mechanisms and energy level assignments in CO<sub>2</sub> lasers we reference Tyte's<sup>23</sup> book. Listed in Appendix I are the allowed infrared emissions from all four laser systems in these ellipsometer instruments.

Switching between the transmitter's four output beams (of unlike wavelength) is computer controlled by shutter pulsing (S1-S4), and translation of various 90° mirror reflectors. The four shutters of Figures 4a-b, one intercepting each laser beam, are open/closed in sequence with and synchronized to the position of three mirrors mounted on a bi-directional translational stage. Either laser beam 1, 2, 3, or 4 traverse the ellipsometer's optical axis (to the scattering sample) by the appropriate mirror position and shutter opening. Figure 4a suggests that just before shutter S4 is switched open (S1-3 are closed), translation stage TS positions to the left where the beam from laser 4 is centered on the outside stage mirror, reflected 90° twice, and sent to the sample. After completion of measurement of all 16 Mueller matrix elements at this wavelength, S4 will close, TS will index to the right, S3 will open, and the next Mueller element set is measured at this new wavelength. (It takes about 30-40 minutes for measuring four spectral sets of 16 matrix elements. Thus, when switching from probe laser 4 to probe laser 3, one has enough time to retune and stabilize laser 4 to a different wavelength. This allows one to make multiple laser lines measurements without halting experimental operations.) Future plans call for incorporating a rapid beam switching system for producing near msec trains of 3- or 4-wavelength pulses in a next generation high-powered frequency agile ellipsometer configuration (Section 6.2). This future field ellipsometer sensor will contain electro-optic variable beam splitter devices, its lasers an order of magnitude more powerful, and its optic collection aperture considerably larger (Figure 4c) via a Cassegrain (or similar construction) telescope.

Optical spectrum analysers are used to visually display the wavelengths of each beam sent to and scattered by the sample. This instrument (SA in Figures 4a-b) consists mainly of a grating (blazed to diffract most efficiently within a certain band of laser transitions) that diffracts a small power percentage of the incident beam to an order whose intensity is displayed on a thermally sensitive florescent (via ultraviolet illumination) screen, darkening in a region aligned to the angle of diffraction. Above the florescent screen, etched markings delineate each CO<sub>2</sub> beam diffractive order to its P- or R-branch transition assignment and band. Two spectrum analysers were used here to cover the extended wavelength range by the isotopic laser emissions.

In latter sections, we discuss the electronic circuit design that regulates power between switching beams incident to the scattering sample. For now, however, notice the linear polarizer mounted to the dc servo-motor (DCSM) driven rotary stage in Figures 4a-b. Its function is to regulate power of each incident beam to a fixed dc PED detector reference voltage, so that operation in the linear region of the MCT detector is maintained (guard against saturation) for all alternating beams incident to the scattering surface. A feedback loop between pyroelectric detector PED and the DCSM accomplishes this regulation function.

#### 4.1.2 Photoelastic Polarization Modulation.

Anti-reflection coated Zinc Selenide (ZnSe) octagonal windows are the active birefringent optical elements that generate polarization modulation in incident laser beams and their collected scattered radiances. These optics are the heart of the ellipsometer: they generate the primary crystal oscillator frequencies and all combination overtones in the spectrum of the intensity waveform measured by the ellipsometer's MCT photoconductive detector. We refer to this complex waveform as the *scattergram*. It encodes the scatterer's Mueller matrix elements. An oscillating birefringence along the crystal's extraordinary (fast) optical axis is produced when resonant periodic compressions/relaxations are applied via a piezoelectric quartz transducer bonded onto its opposite ends. The greater the applied strain along this crystalline plane (within elastic limits), the greater the phase delay in the beams' EM wave component traversing this fast axis, relative to the orthogonal ordinary (slow) field component: along the axis where no phase delay is experienced by the wave during compression and relaxation periods. The net effect: a coherent plane-polarized laser beam incident  $45^\circ$  to the active ZnSe's optical axis (equal fast- and slow-axis EM wave components) becomes polarization modulated (continuous change in Stokes vector with period  $\nu^{-1}$ ,  $\nu$  the transducer frequency) as components recombine on exiting the crystal.

In Figures 4a-c, stacked Ge-plate linear polarizers are positioned before and after the transmitter and receiver PEM's, respectively. They define incident and final Stokes vectors. Before the transmitter PEM, the polarizer has its transmission axis oriented  $45^\circ$  (to fast or slow ZnSe axes) so that beam E-field wave components traversing vertical (fast, extraordinary) and horizontal (slow, ordinary) to the PEM optical axis equate. Birefringence, by nature of ZnSe's lattice structure, imparts a relative delay in phase of the wave component along the crystal's optical extraordinary axis. The piezo-induced pressure along the ZnSe's cleaved ends imparts a changing refractive index along this fast axis – applied strain is proportion to the change of index along the crystal's extraordinary axis and the relative phase difference between components. Vector addition of variably-delayed (fast axis) and unaffected (slow axis) E-field components yields polarization-modulated at the transducer frequency in the beam exiting the PEM. The incident beam modulation from linear to elliptical left- then right-handedness and back to linear has a period of  $\tau_{mod} = \nu_{mod}^{-1} = 29.4 \mu sec$ . In collected scattered radiance, the modulation is  $31.4 \mu sec$ . The variable crystal strain is driven electronically in the PEM's modulator control unit connected to an oscillator circuit located near the modulator head connected to the transducer's bonded quartz plates.

The quarter-wave retardation limit in the ZnSe modulators extends to  $19 \mu m$ , and its clear aperture diameter is 55 mm (wavelength transmission range is  $0.5-19 \mu m$ ). Compression amplitudes in the crystal by the transducer are monitored by computer so that when switching between beams of different wavelength a peak retardation of 137.74 degrees (2.404 rad) is always maintained ( $\delta_0$  in Table 2) by the PEM control unit and oscillator circuit. We later explain in Section 4.3 the importance of maintaining  $\delta_0 = 2.404$  rad between laser beams of unlike wavelengths.

Section 4.4 also shows how to interpret and decode the MCT detected scattergram, i.e., a one-to-one mapping assignments between the primary and overtone intensities in the Fourier transformed scattergrams and the elements of the scattering Mueller matrix elements.

The ellipsometer's transmitter POL-PEM optic unit (Figures 4a-c) polarization-modulates the incident beam Stokes vector at linear driving frequency  $\nu_1 = 33.980$  KHz, while the instrument's receiver PEM-POL unit operates on collected scattered radiance with a modulation frequency of  $\nu_2 = 31.896$  KHz. Together, transmitter and receiver PEM's produce the primary and harmonic combination overtones intensities ( $n\omega_1 \pm k\omega_2$  harmonics) in the Fourier-transformed scattergrams (the MCT detector voltage waveform  $I_f$ ). Modulator frequencies are intentionally offset by 2.064 KHz to insure good separation between primary and overtone frequency components in the scattergram's Fourier power spectrum. The frequencies:  $\nu_{1,2}$ ,  $2\nu_{1,2}$ ,  $\nu_1 + \nu_2$ ,  $2(\nu_1 - \nu_2)$ ,  $2\nu_1 \pm \nu_2$ , and  $\nu_1 + 2\nu_2$ , and the dc component map onto 9 of 16 Mueller elements simultaneously with no intermixing (see Sections 4.2 and 4.3).

#### 4.1.3. The Goniometer Beam Transceiver Arm For In Situ Analyses of Porous Wetted Surfaces.

As mentioned earlier, additional mirror optics were included in one laboratory ellipsometer design so that measurements on porous and granular surfaces can be performed without mechanically handling the scatterer. The goniometer device was fabricated by Mark Schlein of CRDEC and its intended use was to replicate field Mueller matrix measurements in a laboratory environment.

Dry and wetted soil and other terrestrials are analyzed in their normal horizontal position by use of this 'scanning optical projection arm,' as the ejected contaminant naturally diffuses into and spreads across the strata. The scattering in situ measurements is necessary for determining a field feasibility of this technology: the Mueller elements data base by this ellipsometer instrument must bring out properties characteristic of the analyte consistently, and delineate it from scattering by background particles. A typical field scenario is terrain contaminated via an exploded agent round. A 'trained' ellipsometer sensor would probe the suspected contaminated terrain at selective wavelengths and angles to reveal the targeted contaminant's molecular vibrational properties (absorption, VCD, depolarization, etc.). An additional algorithm in that sensor can be further trained to analyze the contaminants fate, such as liquid analyte diffusion, its evaporation and reactivity with the substrate, a changing morphology in crystalline biological compounds, and so on. The information basis to make such analyzes is, again, the analyte-susceptible field of Mueller elements - implying a proper selection of the sensor's incident beam angles and beam wavelengths. A set of systematic experiments may identify patterns in the differential Mueller elements that maximize information on the analyte and discern whether the environment effects its fate.

Since the ellipsometer is a backscattering instrument, the goniometer acts as a transceiver to deliver the beam to the surface, and collect backscattered radiation through some small solid angle limited by the size of the smallest mirror mounted on its arm (its limiting aperture), or some other limiting aperture in the receiver design. In these ellipsometers, the clear aperture of the polarizer mounted to the receiver PEM is the limiting aperture. Consisting of an anodized L-shaped aluminum arm with three IR mirrors mounted at  $45^\circ$  in three corners (yielding three  $90^\circ$  reflections), the goniometer arm is rotated across the upper hemisphere of the sample via a stepper motor rotary stage, computer driven and capable of fractional angle resolution. Details of data acquisition relative to arm movement are provided in Section 4.7.4.

### Pyroelectric (Incident Beam) and Photoconductive (Scattergram) Detectors.

Each laser emission line operates at a separate gain, yielding output beam powers from the ellipsometer's laser sources ranging from 2 Watts or slightly less at the weakest P- or R-branch wing transitions, to a maximum of 15 Watts at the 'hot' mid-branch transition. Gain in the branch transitions are specified according to the grating design (*blazed* to yield high efficiency for certain diffractive orders) in the laser cavity. (The grating is the back resonator optic in the laser head.)

Regulation of intensity in beams incident to the scattering sample guards against MCT detector saturation. The incident beam from each laser, vertical in polarization, is regulated in power by a rotating linear polarizing optic in its path, whose transmission angle is controlled by closed-loop feedback. The electronic feedback in this loop is governed from a circuit that draws its input from the (amplified) output of a pyroelectric detector (PED). That detector monitors a split incident-beam percentage of power (less than 1%) transmitted through the regulation polarizer (Figure 4a-b). The current generated by the PED detector is converted into a voltage, amplified, then compared to a fixed reference voltage that is set to correspond to the maximum power of the weakest laser transition selected in the experimental trial. If voltages of reference and PED output are unequal, then the comparator circuit balances it by feedback to the dc servo motor that constantly adjusts the rotary stage on which the beam power regulation polarizer optic is mounted. Since polarization of the incident beam entering this polarizer is vertical, beam power exiting it (the polarizer is made of Ge stacked-plates oriented at the Brewster angle) is governed by Malus' law:  $I_t = I_i \cos^2\theta$ ; where  $I_i$  is incident beam intensity,  $I_t$  is the exiting beam intensity, and  $\theta(t)$  is the angle the polarizer's transmission axis makes with the vertical plane. (Angle  $\theta(t)$  is under the control of the PED detector output  $\rightarrow$  feedback circuit loop. This circuit adjusts  $\theta$  between switched beams, and regulates power for the duration of beam exposure.) Appendix III illustrates the electronic circuits governing power regulation in more detail.

A Laser Precision Corporation model RS-5900 PED radiometer is used in the power regulation feedback circuitry. It works on the principle that when a photon is absorbed on its Lithium Tantalate ( $\text{LiNbO}_3$ ) photosensitive surface, its temperature rise causes a spontaneous dipole moment increase (the product of induced charge and polarized separation distance in the lattice) that is converted into a voltage. The electronics convert this voltage to beam power (the detector is equipped with a 1.5 Hz chopper) density the company claims is traceable to NIST standards. Sensitivity in the feedback loop from this detector output is expected to regulate incident beam power to one percent or better of the set reference voltage value.

Light scattered by the sample is detected by a liquid nitrogen cooled Mercury-Cadmium-Telluride (MCT,  $\text{Hg}_{1-x}\text{Cd}_x\text{Te}$ ) photoconductive chip. Since throughput (a measure of scattered laser power collected through the instrument's optical system over some solid angle projected by the receiver MCT chip) is small in these ellipsometer systems, the high detectivity of cryogenic MCT detection technology is required for acquiring scattergrams from the irradiation zone with fast enough response time. (Scattering from most samples of interest is near Lambertian, i.e., isotropic radiance. Also, rough dielectric surfaces are generally good absorbers of the IR beam energy, typical of most terrains.) The responsivity of MCT detectors provide sub microsecond time constants, good enough resolution for capturing the scattergram signal and making the Mueller mappings with  $\leq 10\%$  error. The MCT detector response time allows accurate digital recording of the scattergram with satisfactory resolution for phase extractions. (The Fourier power spectrum of the scattergram must clearly demarcates all PEM harmonics up to 100 KHz.)

MCT photoconductive detectors operate on a principle different from PED's. Scattered photons from the sample strike the MCT semiconductor chip generating free carrier electron-hole pairs producing changes in surface electrical resistance. Liquid nitrogen cooling of the chip's surface is required for reducing background noise and increasing detectivity of the photon-induced electron-hole pairs. The ellipsometer's scattergram represents a changing surface resistance of the semiconductor via intensity variation in the focused scattered beam

radiance illuminating the chip. The varying surface resistivity is converted into a voltage change (the scattergram) by applying a bias current through the MCT chip. Consequently, the scattergram is preamplified and then sent to a variable gain (voltage controlled) amplifier, filtered and digitized (Section 6.1), or distributed to the phase sensitive detection analog electronics (Section 4.5).

MCT specifications for the ellipsometer systems are: (1) a cutoff wavelength to at least the maximum wavelength output of the  $C^{14}O_2^{16}$  laser; (2) a time constant small enough to capture, with good resolution, the real time transient waveform containing primary and harmonic frequencies to 100 KHz; and (3) a detectivity high enough so that, at grazing incident beam angle, signal-to-noise ratio is greater than one. These are met or are exceeded by a MCT detector we have purchased from EG&G Judson. That specific detector has the respective properties:  $12.4 \mu m$ ;  $0.5 \mu sec$ ; and  $2.0-5.1 \times 10^{10} cm Hz^{0.5} W^{-1}$ . Surface area of the MCT chip is  $1 mm^2$ , thus focusing/demagnifying the scattered radiance with lens L onto the detector chip is required to increase intensity  $I_f$  to detectable levels in the scattergram.

#### 4.2 Ellipsometer System and Sub-System Matrices.

Signal definition and fundamentals of data processing in these ellipsometer systems are now discussed. Recall from Sections 2.1 and 2.2 the principals of Stokes vector and Mueller matrix: the incident Stokes vector ( $s^i$ ) is altered upon scattering ( $s^f$ ) interactions according to the Mueller matrix (F) transformation. Consider the sequence of optical elements in the ellipsometer system of Figure 4a, starting with the transmitter POL-PEM (the linear polarizer mated to this PEM defines the initial Stokes vector components  $s_j^i$ ;  $j=0,1,2,3$ ) and ending with the receiver PEM-POL (the linear polarizer mated to this PEM defines the final Stokes vector components  $s_j^f$ .)

Each optical element positioned between POL's defining  $s^{i,f}$  are classified by their separate Mueller matrix operator, and thus the transformed Stokes vector through the goniometer-type ellipsometer system is given by:

$$s^f = F s^i \quad (5)$$

$$F = P_r G_r M_3 R M_2 M_3 F M_1 M_2 M_1 R G_t P_t \quad (6a)$$

where subscripts t and r denote transmitter and receiver paths; P and G are matrices of polarizer and PEM optics, respectively;  $M_1$ ,  $M_2$ , and  $M_3$  are matrices of the flat  $45^\circ$  mirrors mounted on the goniometer arm; F is the matrix of the scatterer, the unknown of primary importance; and R is a goniometer rotation operator. The product of F matrices do not commute, and must appear in proper order from transmitter through receiver optics.

Before proceeding, we should clarify the function of R. Rotation operator R changes the reference measurement plane in which the Stokes vectors are defined ( $\epsilon_1, \epsilon_2$  in Equations 1-3). In Figures 4a-c, the optic platform is the reference plane defining the basis unit vectors  $\hat{\epsilon}_1$  and  $\hat{\epsilon}_2$ . If the goniometer arm is positioned at an angle other than  $\pm 90^\circ$ , the Stokes vector becomes referenced in its new goniometer reference plane defined by the new basis unit vectors  $\hat{\epsilon}_1', \hat{\epsilon}_2'$  (the plane defined by beam reflections by the three goniometer mirrors), rotated from the old reference plane by polar angle  $\theta$ . The R operator third from the end in Equation 6a transforms the Stokes vector from its old reference plane  $\hat{\epsilon}_1, \hat{\epsilon}_2$  to the new goniometer

reference plane. Therefore,  $M_{1,2,3}$  and  $F$  (Table 2) are now referenced in this new frame. On exiting the goniometer transceiver, scattered light is operated on by  $R$  again; forth matrix from the left hand side of Equation 6a, which transforms the Stokes vector back to the old  $\hat{e}_1, \hat{e}_2$  reference frame. Positioning of  $R$  operators in Equation 6 must follow the precise entrance and exit points of the goniometer transceiver.

A new matrix product is defined below, which we shall call the ellipsometer's system matrix.

$$\Psi = M_4 R M_1 M_2 M_3 F M_3 M_2 M_1 R \quad (6b)$$

By substituting Equation 6b into 6a, the sample matrix can now be rewritten as:

$$F = P_r G_r \Psi G_t P_t. \quad (6c)$$

#### 4.3 MCT Detector Waveform of Scattered Radiation.

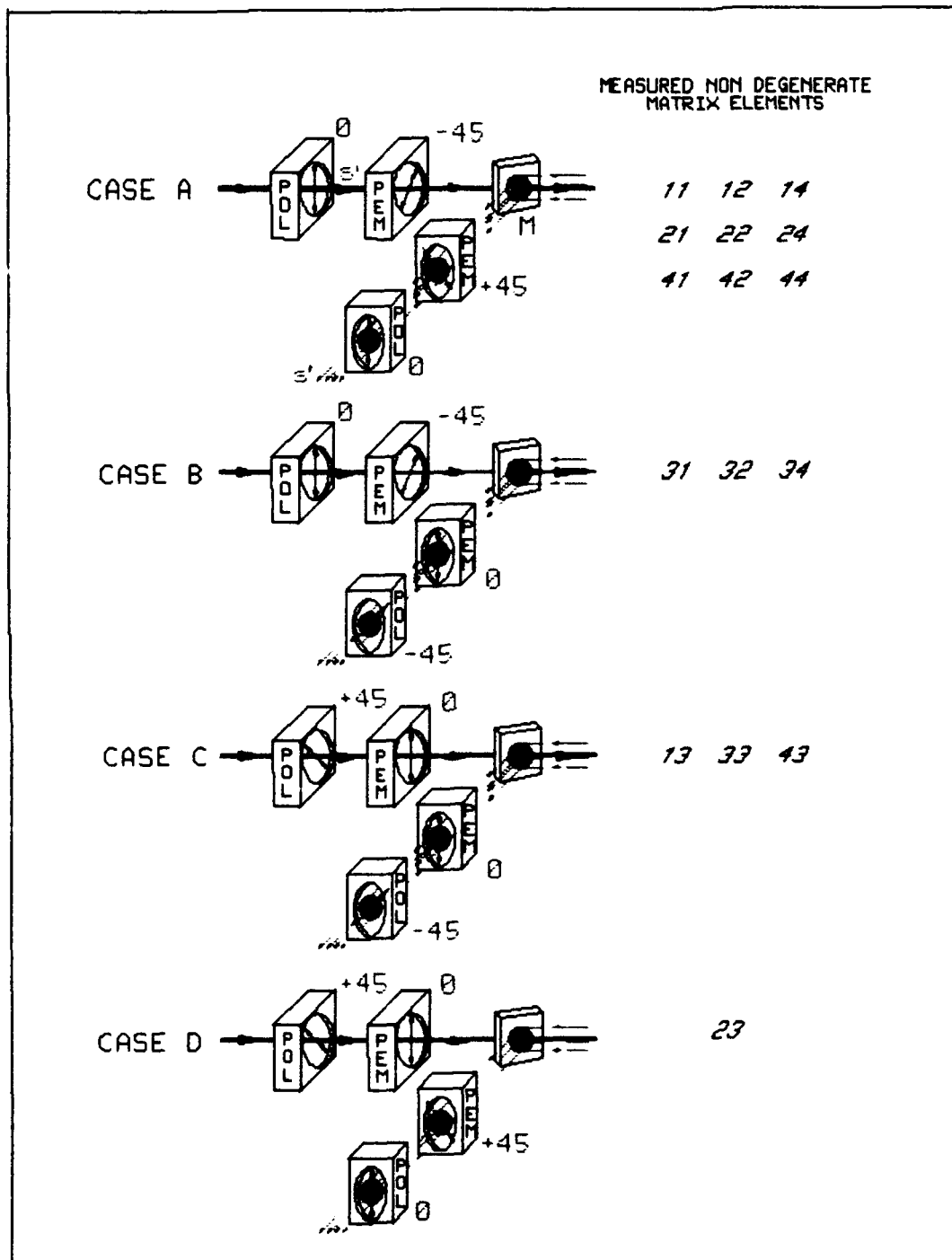
We proceed in deriving a functional form of the MCT scattergram generated by the 2-modulator ellipsometer systems of Figure 4a-c. Four optical permutations of each POL-PEM pair are illustrated in Figure 5 as defined by Cases A, B, C and D.

##### Case A. Vertical $-45^\circ$ : $+45^\circ$ Vertical

In the notation used above, axes of polarizer and modulator optics in *transmitter* and *receiver* paths of the ellipsometers are separated by a colon. The left part, *Vertical  $-45^\circ$* , denotes the *receiver* part of this instrument's POL-PEM unit, with polarizer (Vertical) transmission axis followed by its attached PEM retarder (fast) axis ( $-45^\circ$ ). The second entries,  *$+45^\circ$  Vertical* to the right of the colon in Case A, denotes the *transmitter* section PEM-POL unit with retarder axis ( $+45^\circ$ ) to the left and attached polarizer axis (Vertical) to the right. Angles are measured relative to the plane of the optics table (the reference plane), hence Vertical is  $0^\circ$  or perpendicular to the optical platform, and  $+(-)45^\circ$  is measured in the clockwise (counter clockwise) direction one-eighth revolution as viewing each PEM from the laser source.

Figure 5 illustrates four optic orientations in the ellipsometer. First consider the configuration given by Case A. Polarizers mated to each modulator set the incident and final Stokes vectors accordingly;

$$s^i = I_0 \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}, \quad s^f = I_f \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}. \quad (7)$$



**Figure 5.** The optical orientations of polarizer-modulator crystal axis producing the system Mueller matrix elements, as measured from the ellipsometer's scattergram by the analog data collection unit. Note that the Mueller element contributions by mirror M can be compensated by rearrangement and insertion of an identical mirror. (See Section 4.6.3.)

Substitution of Equations 7a, 6c, and the appropriate matrices of Table 2 into Equation 5, then left-multiplication by  $s^f$  results in the following expression for scattering intensity exiting the receiver polarizer and incident to the MCT detector chip.

$$I_f = \frac{I_0}{8} (1 \ -1 \ 0 \ 0) \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\delta_0 \cos \omega_2 t) & 0 & +\sin(\delta_0 \cos \omega_2 t) \\ 0 & 0 & 1 & 0 \\ 0 & -\sin(\delta_0 \cos \omega_2 t) & 0 & \cos(\delta_0 \cos \omega_2 t) \end{pmatrix} \quad (8a)$$

$$\begin{pmatrix} \psi_{11} & \psi_{12} & \psi_{13} & \psi_{14} \\ \psi_{21} & \psi_{22} & \psi_{23} & \psi_{24} \\ \psi_{31} & \psi_{32} & \psi_{33} & \psi_{34} \\ \psi_{41} & \psi_{42} & \psi_{43} & \psi_{44} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\delta_0 \cos \omega_1 t) & 0 & -\sin(\delta_0 \cos \omega_1 t) \\ 0 & 0 & 1 & 0 \\ 0 & +\sin(\delta_0 \cos \omega_1 t) & 0 & \cos(\delta_0 \cos \omega_1 t) \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{aligned} &= \frac{I_0}{2} (\psi_{11} - \psi_{12} \cos(\delta_0 \cos \omega_1 t) - \psi_{14} \sin(\delta_0 \cos \omega_1 t) - \psi_{21} \cos(\delta_0 \cos \omega_2 t) \\ &+ \psi_{22} \cos(\delta_0 \cos \omega_2 t) \cos(\delta_0 \cos \omega_1 t) + \psi_{24} \cos(\delta_0 \cos \omega_2 t) \sin(\delta_0 \cos \omega_1 t) \\ &- \psi_{41} \sin(\delta_0 \cos \omega_2 t) + \psi_{42} \sin(\delta_0 \cos \omega_2 t) \cos(\delta_0 \cos \omega_1 t) \\ &+ \psi_{44} \sin(\delta_0 \cos \omega_2 t) \sin(\delta_0 \cos \omega_1 t)) \end{aligned} \quad (8b)$$

The maximum retardation amplitude along the modulator's fast axis is  $\delta_0$ . The driving resonance crystal frequencies in receiver and transmitter PEM's are, respectively,  $\omega_1$  and  $\omega_2$ . The  $\delta_0$  and  $\omega$  variables in arguments of the sine and cosine terms of Equation 8b can be separated by substitution of the Bessel generating function<sup>24</sup>:

$$e^{i\delta_0 \cos(\omega t)} = J_0(\delta_0) + 2 \sum_{k=1}^{\infty} i^k J_k(\delta_0) \cos k \omega t \quad (9)$$

where  $J_k$  are Bessel functions of  $k^{\text{th}}$  order, and  $i$  is the imaginary number  $\sqrt{-1}$ . Both PEM transducers are set to yield  $\delta_0 = 2.404$  radians (in the arguments of the Bessel functions). This nulls the zero order Bessel function:  $J_0(2.404) = 0$ . The real and imaginary components of Equation 9 can now be expressed by the following infinite series expansions.

$$\frac{\cos(\delta_0 \cos \omega t)}{2} = -J_2(\delta_0) \cos 2\omega t + J_4(\delta_0) \cos 4\omega t - J_6(\delta_0) \cos 6\omega t + \dots \quad (10a)$$

$$\frac{\sin(\delta_0 \cos \omega t)}{2} = J_1(\delta_0) \cos \omega t - J_3(\delta_0) \cos 3\omega t + J_5(\delta_0) \cos 5\omega t - \dots \quad (10b)$$

By substituting Equations 10a-b into Equation 8b and factoring, the amplitude and frequency components can be separated to yield the scattergram intensity waveform expansion  $I_f$ .

$$\begin{aligned} \frac{I_f}{I_0} = & \dots + \frac{1}{2} [J_3^2(\delta_0) \cos(3\omega_2 t) \cos(3\omega_1 t) - J_1(\delta_0) J_3(\delta_0) \cos(\omega_2 t) \cos(3\omega_1 t) \\ & - J_1(\delta_0) J_3(\delta_0) \cos(3\omega_2 t) \cos(\omega_1 t) + J_1^2(\delta_0) \cos(\omega_2 t) \cos(\omega_1 t)] \psi_{44} \\ & - \frac{1}{2} [J_3(\delta_0) J_4(\delta_0) \cos(3\omega_2 t) \cos(4\omega_1 t) - J_1(\delta_0) J_4(\delta_0) \cos(\omega_2 t) \cos(4\omega_1 t) \\ & - J_1(\delta_0) J_3(\delta_0) \cos(3\omega_2 t) \cos(2\omega_1 t) + J_1(\delta_0) J_2(\delta_0) \cos(\omega_2 t) \cos(2\omega_1 t)] \psi_{42} \\ & + [J_3(\delta_0) \cos(3\omega_2 t) - J_1(\delta_0) \cos(\omega_2 t)] \psi_{41} + \frac{1}{2} [J_3(\delta_0) J_4(\delta_0) \cos(4\omega_2 t) \cos(3\omega_1 t) \\ & - J_2(\delta_0) J_3(\delta_0) \cos(2\omega_2 t) \cos(3\omega_1 t) - J_1(\delta_0) J_4(\delta_0) \cos(4\omega_2 t) \cos(\omega_1 t) \\ & + J_1(\delta_0) J_2(\delta_0) \cos(2\omega_2 t) \cos(\omega_1 t)] \psi_{24} + \frac{1}{2} [J_4^2(\delta_0) \cos(4\omega_2 t) \cos(4\omega_1 t) \\ & - J_2(\delta_0) J_4(\delta_0) \cos(2\omega_2 t) \cos(4\omega_1 t) - J_2(\delta_0) J_4(\delta_0) \cos(4\omega_2 t) \cos(2\omega_1 t) \\ & + J_2^2(\delta_0) \cos(2\omega_2 t) \cos(2\omega_1 t)] \psi_{22} - [J_4(\delta_0) \cos(4\omega_2 t) - J_2(\delta_0) \cos(2\omega_2 t)] \psi_{21} \\ & - [J_4(\delta_0) \cos(4\omega_1 t) - J_2(\delta_0) \cos(2\omega_1 t)] \psi_{12} + [J_3(\delta_0) \cos(3\omega_1 t) \\ & - J_1(\delta_0) \cos(\omega_1 t)] \psi_{14} + \psi_{11} \end{aligned} \quad (11)$$

Equation 11 consists of a dc component which is always matrix element  $\psi_{11}$ , and an infinite number of overtones of the modulator driving frequencies that diminish in amplitude as the order and product of higher-order Bessel function coefficients increase. We truncate the series of Equation 11 for order  $k \geq 3$  (a good approximation of the scattergram given the S/N of these instruments) and correlate the remaining nine strongest Fourier intensities one-to-one to their respective Mueller system elements  $\psi_{ij}$  (in the coefficients of Equation 11). The

transducer primary frequencies for each PEM were chosen as  $\omega_1/2\pi = 31.896$  KHz in the transmitter crystal, and  $\omega_2/2\pi = 33.960$  KHz in the receiver crystal. Furthermore, with these choices of resonant PEM driving frequencies, spectral intensities in  $I_f$  are separated by a minimum of 2.064 KHz. Thus, standard lock-in amplifiers can quickly detect and phase-match each scattergram frequency component to its reference primary or strongest overtone modulator frequency without interference from a neighboring harmonic.

The analog data acquisition system separately conducts the dc scattergram component ( $\psi_{11}$ ) to a separate analog-to-digital (A/D) converter channel in one unit module (Appendix III). For the ac signals (elements other than  $\psi_{11}$ ), eight phase-sensitive detector circuit cards are set to the reference modulator frequencies of Equation 8b. These lock-in amplifiers electronically multiply their respective reference modulator frequency and real-time scattergram waveforms. The result is an analog output that represents the phase difference between scattergram filtered to the frequency of a reference modulator waveform. The 8 ac Mueller data channels are received simultaneously in another module of the acquisition unit, and all 9 channels are conducted to an A/D converter that strobes through each channel, recording its output in an appropriate file of CPU memory.

In terms of  $\frac{I_f}{I_0}$ , Equation 11, the most intense primary and overtone frequency components of the scattergram are:

$$\begin{aligned} \frac{I_f}{I_0} = & \dots + 0.270\psi_{44}\cos(\omega_2 \pm \omega_1)t - 0.224\psi_{42}\cos(\omega_2 \pm 2\omega_1)t - 0.520\psi_{41}\cos(\omega_2 t) \quad (12) \\ & - 0.224\psi_{24}\cos(2\omega_2 \pm \omega_1)t + 0.186\psi_{22}\cos(2\omega_2 \pm 2\omega_1)t + 0.431\psi_{21}\cos(2\omega_2 t) \\ & + 0.431\psi_{12}\cos(2\omega_1 t) - 0.520\psi_{14}\cos(\omega_1 t) + \psi_{11} \end{aligned}$$

where  $\cos(A \pm B) = \cos(A+B) + \cos(A-B)$ . The sign before each coefficient indicates a relative  $\pi$ -phase difference between elements, i.e., in Equation 12  $\psi_{24}$  and  $\psi_{42}$  are  $180^\circ$  out of phase with each other.

#### Case B. Vertical $-45^\circ$ : Vertical $-45^\circ$

After the six Mueller elements of Case A are measured as functions of backscattering angle and wavelength, the ellipsometer rotates the receiver PEM-POL pair  $-45^\circ$  (as viewed from the laser source) producing the new orientation called Case B, Figure 5. The new incident and scattering Stokes vectors are:

$$\mathbf{s}^i = I_0 \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{s}^f = I_f \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \end{pmatrix}. \quad (13)$$

\* Equation 11 is truncated for Bessel terms greater than 3<sup>rd</sup> order. Recall that  $\delta_0 = 2.404$  rad. The law of cosines was used in deriving Equation 12, as were these tabulated values:  $J_1(\delta_0) = 0.520$ ,  $J_2(\delta_0) = 0.431$ .

Substituting matrices from Tables 1 and 2 produces the following analogous expression to Equation 8.

$$\frac{I_f}{I_0} = \frac{1}{8}(1 \ 0 \ -1 \ 0) \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\delta_0 \cos \omega_2 t) & -\sin(\delta_0 \cos \omega_2 t) \\ 0 & 0 & +\sin(\delta_0 \cos \omega_2 t) & \cos(\delta_0 \cos \omega_2 t) \end{pmatrix} \quad (14a)$$

$$\begin{pmatrix} \psi_{11} & \psi_{12} & \psi_{13} & \psi_{14} \\ \psi_{21} & \psi_{22} & \psi_{23} & \psi_{24} \\ \psi_{31} & \psi_{32} & \psi_{33} & \psi_{34} \\ \psi_{41} & \psi_{42} & \psi_{43} & \psi_{44} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\delta_0 \cos \omega_1 t) & 0 & -\sin(\delta_0 \cos \omega_1 t) \\ 0 & 0 & 1 & 0 \\ 0 & +\sin(\delta_0 \cos \omega_1 t) & 0 & \cos(\delta_0 \cos \omega_1 t) \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

$$= \dots + 0.270\psi_{44}\cos(\omega_2 \pm \omega_1)t + 0.224\psi_{42}\cos(\omega_2 \pm 2\omega_1)t - 0.520\psi_{41}\cos(\omega_2 t) \quad (14b)$$

$$+ 0.224\psi_{34}\cos(2\omega_2 \pm \omega_1)t + 0.186\psi_{32}\cos(2\omega_2 \pm 2\omega_1)t + 0.431\psi_{31}\cos(2\omega_2 t)$$

$$+ 0.431\psi_{12}\cos(2\omega_1 t) - 0.520\psi_{14}\cos(\omega_1 t) + \psi_{11}$$

Notice that three new elements appear in the above expression:  $\psi_{31}$ ,  $\psi_{32}$ , and  $\psi_{34}$ .

Case C.  $+45^\circ$  Vertical:Vertical  $-45^\circ$

By this time, 12 of 16 Mueller elements have been measured. The ellipsometer will now send an index command to the motion controller, causing the transmitter POL-PEM to rotate precisely  $+45^\circ$  for producing Case C, Figure 5. Incident and scattered Stokes vectors become:

$$\mathbf{s}^i = I_0 \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{s}^f = I_f \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \end{pmatrix}, \quad (15)$$

and detector intensity is represented by the product of the following six matrices. (See Tables 1 and 2.)

$$\frac{I_f}{I_0} = \frac{1}{8}(1 \ 0 \ -1 \ 0) \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\delta_0 \cos \omega_2 t) & -\sin(\delta_0 \cos \omega_2 t) \\ 0 & 0 & +\sin(\delta_0 \cos \omega_2 t) & \cos(\delta_0 \cos \omega_2 t) \end{pmatrix} \quad (16a)$$

$$\begin{pmatrix} \psi_{11} & \psi_{12} & \psi_{13} & \psi_{14} \\ \psi_{21} & \psi_{22} & \psi_{23} & \psi_{24} \\ \psi_{31} & \psi_{32} & \psi_{33} & \psi_{34} \\ \psi_{41} & \psi_{42} & \psi_{43} & \psi_{44} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\delta_0 \cos \omega_1 t) & -\sin(\delta_0 \cos \omega_1 t) \\ 0 & 0 & +\sin(\delta_0 \cos \omega_1 t) & \cos(\delta_0 \cos \omega_1 t) \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\begin{aligned} &= \dots + 0.270\psi_{44}\cos(\omega_2 \pm \omega_1)t - 0.224\psi_{43}\cos(\omega_2 \pm 2\omega_1)t - 0.520\psi_{41}\cos(\omega_2 t) \\ &+ 0.224\psi_{34}\cos(2\omega_2 \pm \omega_1)t - 0.186\psi_{33}\cos(2\omega_2 \pm 2\omega_1)t + 0.431\psi_{31}\cos(2\omega_2 t) \\ &- 0.431\psi_{13}\cos(2\omega_1 t) - 0.520\psi_{14}\cos(\omega_1 t) + \psi_{11} \end{aligned} \quad (16b)$$

Again, three of the nine Mueller matrix elements in the above expression are new mappings according to new polarizer and modulator optic orientations. They are:  $\psi_{13}$ ,  $\psi_{33}$ , and  $\psi_{43}$ .

*Case D. +45° Vertical: +45° Vertical*

By this time, 15 of 16 Mueller elements over angle and wavelength to the scattering sample have been measured. The software module that controls the system's optical permutation of axes will now cause a +45° rotation of the receiver PEM-POL, producing the final Case D in Figure 5. Incident and scattered Stokes vectors are now:

$$\mathbf{s}^i = I_0 \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{s}^f = I_f \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}, \quad (17)$$

and the detector intensity is represented by the product of these seven matrices. (Consult Tables 1 and 2.)

$$\frac{I_f}{I_0} = \frac{1}{8} (1 \ -1 \ 0 \ 0) \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\delta_0 \cos \omega_2 t) & 0 & +\sin(\delta_0 \cos \omega_2 t) \\ 0 & 0 & 1 & 0 \\ 0 & -\sin(\delta_0 \cos \omega_2 t) & 0 & \cos(\delta_0 \cos \omega_2 t) \end{pmatrix}. \quad (18a)$$

$$\begin{pmatrix} \psi_{11} & \psi_{12} & \psi_{13} & \psi_{14} \\ \psi_{21} & \psi_{22} & \psi_{23} & \psi_{24} \\ \psi_{31} & \psi_{32} & \psi_{33} & \psi_{34} \\ \psi_{41} & \psi_{42} & \psi_{43} & \psi_{44} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\delta_0 \cos \omega_1 t) & -\sin(\delta_0 \cos \omega_1 t) \\ 0 & 0 & +\sin(\delta_0 \cos \omega_1 t) & \cos(\delta_0 \cos \omega_1 t) \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\begin{aligned}
&= \dots - 0.270\psi_{44}\cos(\omega_2 \pm \omega_1)t + 0.224\psi_{43}\cos(\omega_2 \pm 2\omega_1)t + 0.520\psi_{41}\cos(\omega_2 t) \quad (18b) \\
&+ 0.224\psi_{24}\cos(2\omega_2 \pm \omega_1)t - 0.186\psi_{23}\cos(2\omega_2 \pm 2\omega_1)t + 0.431\psi_{21}\cos(2\omega_2 t) \\
&- 0.431\psi_{13}\cos(2\omega_1 t) - 0.520\psi_{14}\cos(\omega_1 t) + \psi_{11}
\end{aligned}$$

The new and final 16<sup>th</sup> element in the above equation that completes the Mueller matrix field of mappings is  $\psi_{23}$ . It takes  $\approx 4$  s for the transceiver POL-PEM optics to cycle through Cases A-D per beam wavelength per backscattering angle.

In the ellipsometer configuration of Figure 4b, analysis of the sample scattergram is considerably less complex than from that of configurations Figure 4a or Figure 4c. All M, and R operators are now the unit matrix. The preceding equations still apply, except now the system matrix is the sample matrix, i.e.,  $\psi_{ij} = f_{ij}$ , since only the scattering sample lies between linear polarizers defining incident and final Stokes vectors. Sample matrix elements are thereby directly obtained at Fourier intensities of the MCT detector  $I_f$  waveform.

#### 4.4 Lock-in Detection Matrix of Primary and Overtone PEM Modulator Frequencies.

The discussion in the previous section on detector waveform production is summarized here by inclusion of the important *frequency matrix*, assigning primary and overtones appearing in the MCT intensities of Equations 12, 14b, 16b, and 18b to Mueller system matrix elements according to the four PEM-POL optic configurations cited as Case A through D.

Table 3 contains important information on the ellipsometer's sequence of electronic signal acquisitions. Computer automation at startup initializes the system to Case A in Figure 5. All eight lock-in amplification channels in the analog detector unit (see Appendix III) plus two dc channels (before and after automatic gain operation) are producing outputs that are digitized and stored in CPU memory. These data files are preprocessed and organized into Mueller element files with header blocks containing information on backscattering angles, beam wavelengths, type scatterer, sample surface topography (mean squared heights and slopes), type analyte, contaminant mass density, irradiation time, and other parameters that depend on experiment measurement options.

Consider again the sequence of steps for data collection and storage. The ellipsometer operation starts with measurements of the initial nine Mueller element channels of Case A. A change of optical orientation produces Case B, and the system proceeds with measurements of the next 3 nondegenerate matrix element channels. The A/D converter board recognizes and activates these three channels corresponding to the new Mueller elements, and software appropriately routes the data to three new files. Twelve of sixteen elements have now been acquired and stored in memory. The six channels that are degenerate in Case B, i.e., the duplicated Mueller elements collected from Case A, may be deactivated for efficient use of CPU memory. (All channels are checked in calibration experiments to assure repeatability between degenerate elements from one optical orientation to the next.) The computer next sends an ANSI code to the controller and Case C is produced. Again, three of nine channels contain nondegenerate elements and are active, bringing the number of acquired and stored matrix elements to fifteen. The sixteenth element is acquired after the computer produces the final configuration shown as Case D in Figure 5, viz, one nondegenerate element, one required active channel for collection and storage of data. Therefore, for a complete Mueller matrix measurement per independent experimental variable, the Amplitude and Phase Sensitive Detector (APSD) activates its data channels in sequence 9:3:3:1 corresponding to optic permutations labeled Cases A-D.

Table 3. Lock-in frequencies for phase sensitive detection, and corresponding optical alignments for measurement of the full Mueller matrix. Grouped in part (a) are the primary and major combination frequency components of transmitter ( $\omega_1$ ) and receiver ( $\omega_2$ ) photoelastic modulator's that map onto each Mueller matrix element from the scattergram's Fourier intensities. A parenthetical number next to each frequency implies that component's relative strength compared to component dc=1. The entries for each matrix element are a result of the orientations of linear polarizer and modulator optical axes as grouped in part (b). (See Figure 5.)

SELECT PRIMARY MUELLER MATRIX LOCK-IN FREQUENCIES (KHz)			
$\psi_{11}$ $dc$ $dc$ $dc$ $dc$	$\psi_{12}$ $2\omega_1$ (0.431) $2\omega_1$ (0.431) $0$ $0$	$\psi_{13}$ $0$ $0$ $2\omega_1$ (0.431) $2\omega_1$ (0.431)	$\psi_{14}$ $\omega_1$ (0.520) $\omega_1$ (0.520) $\omega_1$ (0.520) $\omega_1$ (0.520)
$\psi_{21}$ $2\omega_2$ (0.431) $0$ $0$ $2\omega_2$ (0.431)	$\psi_{22}$ $2\omega_2-2\omega_1$ (0.186) $0$ $0$ $0$	$\psi_{23}$ $0$ $0$ $0$ $2\omega_2-2\omega_1$ (0.186)	$\psi_{24}$ $\omega_1+2\omega_2$ (0.224) $0$ $0$ $\omega_1+2\omega_2$ (0.224)
$\psi_{31}$ $0$ $2\omega_2$ (0.431) $2\omega_2$ (0.431) $0$	$\psi_{32}$ $0$ $2\omega_2-2\omega_1$ (0.186) $0$ $0$	$\psi_{33}$ $0$ $0$ $2\omega_2-2\omega_1$ (0.186) $0$	$\psi_{34}$ $0$ $\omega_1+2\omega_2$ (0.224) $\omega_1+2\omega_2$ (0.224) $0$
$\psi_{41}$ $\omega_2$ (0.520) $\omega_2$ (0.520) $\omega_2$ (0.520) $\omega_2$ (0.520)	$\psi_{42}$ $2\omega_1+\omega_2$ (0.224) $2\omega_1+\omega_2$ (0.224) $0$ $0$	$\psi_{43}$ $0$ $0$ $2\omega_1+\omega_2$ (0.224) $2\omega_1+\omega_2$ (0.224)	$\psi_{44}$ $\omega_2+\omega_1$ (0.270) $\omega_2+\omega_1$ (0.270) $\omega_2+\omega_1$ (0.270) $\omega_2+\omega_1$ (0.270)

(a)

ANGLES RELATIVE TO REFERENCE PLANE NORMAL			
TRANSMITTER		RECEIVER	
LINEAR POLARIZER	PHASE MODULATOR	PHASE MODULATOR	LINEAR POLARIZER
Vertical	-45°	+45°	Vertical
Vertical	-45°	Vertical	-45°
+45°	Vertical	Vertical	-45°
+45°	Vertical	+45°	Vertical

(b)

#### 4.5 Analog Amplitude and Phase Sensitive Detection Electronics.

This section presents the assembly of electronic circuits and modules of the ellipsometer analog data acquisition system. We denote the analog circuitry presented here as 'first generation.' Moreover, we now are pursuing development of 'second generation' real time digital data acquisition systems (Section 6.1) that will be compared directly, regarding performance and economy, to this 9-channel analog unit. Furthermore, we have already begun development of a neural network architecture that will eventually interface to the analog port output module, or the output connector of the digital acquisition unit (see Section 6.3). This is designated a 'third generation' phase sensitive detection.

The modular analog signal acquisition unit is now, however, in an advanced engineering stage and the first tested in all ellipsometer configurations. Figure 6 is a basic overlay of how information in the scattergram is mapped into the Mueller elements, showing major frequency synthesizer, APSD channels, and software matrix normalization interfaces. In Appendix III, a more detailed electronic breakdown of the complete APSD unit is provided.

##### 4.5.1 Mueller Matrix Acquisition: Theory of Operation.

The hardware of the analog detection system used to collect and separate various amplitude and phase informations from the detector waveform (Section 4.3) will now be discussed. In constructing a finite set of Mueller matrices, this detector processes the input waveform with eight discrete modules. The basic function of these modules as a unit is to collect and separate the pre-amplified MCT scattergram into nine discrete frequencies (Table 3), and from these generate a scalar (number) which corresponds to the cosine of the difference in phase between these frequencies and the phase of nine respective reference frequencies derived from the two photoelastic modulation oscillator circuits. These resulting scalars are accessed via a RS-232C port that integrates signal processor and host computer mainframe.



#### 4.5.2 Reference Frequencies Generator, Mueller Elements Lock-in Amplifications.

The entire signal processing element of the analog Mueller matrix acquisition unit essentially contains a signal reference generator and a signal comparator. The signal reference generator consists of Modules I, II, and III (see Figure AIII.7a). Module I accepts the transistor-transistor logic (TTL) frequencies  $\omega_1$ ,  $2\omega_1$  and  $\omega_2$ ,  $2\omega_2$  direct from the PEM oscillators, and synthesizes four sinusoid waveforms ( $3.0-4.0 V_{p-p}$ ) each at a frequency and phase relative to the trigger pulses from the corresponding modulator. These four sinusoidal waveforms are then multiplied by Module II to produce the overtones:  $\omega_1+\omega_2$  (65.86 KHz),  $\omega_1+2\omega_2$  (97.75 KHz),  $2\omega_1+\omega_2$  (99.82 KHz), and  $2\omega_1-2\omega_2$  (4.13 KHz). The multiplier board of Module II includes buffering and harmonic filter circuits for the four PEM sinusoid waveform inputs from Module I. The output of Module II thus consists of eight sinusoid waveforms,  $2 V_{p-p}$ , including primary frequencies  $\omega_1$  (33.96 KHz),  $2\omega_1$ (67.92 KHz),  $\omega_2$ (31.90 KHz),  $2\omega_2$ (63.79 KHz) and the overtones (Table 3).

Note, that the four overtone product sinusoids consist of both sum and difference values of the multiplied input primary frequencies. We have chosen the specific overtones indicated in Table 3 because they correspond to the eight most intense Fourier amplitudes (greatest signal-to-noise ratio among Mueller components) obtained from the detected scattergram.

Module III conditions the product waveforms (from Module II) by selective bandpass filtering of the desired overtone frequencies, and provides an adjustable phase shift to all eight (primary plus overtones) reference waveforms. These reference waveforms can be obtained through BNC connectors  $J_{36} - J_{43}$  ( $2.0-3.0 V_{p-p}$  at  $Z=50 \Omega$ ) for calibration purposes.

Module IV (see Figure AIII.8) consists of the phase sensitive detector (PSD) boards. The eight reference frequencies synthesized from Modules I-III are inputs to the individual PSD circuit cards tuned for that frequency. The PSD circuits multiply (dot product) reference and MCT detector waveforms. With reference (amplitude A) and MCT scattergram (amplitude B) waveforms connect to the input channel of each PSD board, an analog dc output voltage is produced with magnitude proportional to the cosine of the phase angle between input waveforms and amplitude product:  $AB\cos(\theta_a-\theta_b)$ . All PSD outputs are buffered with gain control potentiometers ( $R_{17} - R_{24}$ ) located on the unit's front panel, also for calibration purposes. These buffered signals are externally available through eight  $50\Omega$  BNC terminators ( $J_9 - J_{16}$ ).

#### 4.5.3 Digitization of the APSD Outputs (Mueller Matrix Channels).

Module V (see Figure AIII.8) of the analog phase sensitive detector unit consists of a microprocessor controlled model ST701 Analog-to-Digital converter (ADC) manufactured by DATEL, Incorporated. With a 12/20 Intel compatible VME plug-in board, this unit is used as a stand alone processing system that is accessible through the breadboard's RS232 serial link, and controlled through commands issued by the host microvax computer. Among its other tasks, the host computer strobes the ADC for acquisition of all nine analog APSD channels of data synchronous to experimental variable(s), and options of measurements coded in the system control software package (Appendix IV).

#### 4.5.4 Servo Loops for Variable MCT Gain Control and Incident Beam Power Regulation.

Module VI of the analog APSD consists of an automatic-gain-control (AGC) amplifier, fabricated for this system by Analog Modules, Inc., that is connected directly to the ellipsometer's MCT detector output port. The feedback function in the AGC amplifier controls current through the detector's split dc load resistance, to maintain a constant pre-set dc amplitude in the MCT output for all variations in the experimental parameters, including laser wavelength and backscattering angle. As these independent variables are controlled by the instrument's automation software, the AGC regulates the dc scattergram component, thereby causes normalization of all ac matrix components. (Division of phase Mueller elements by the dc element  $f_{11}$ . All elements except the dc element are bounded between +1 and -1.)

There is a provision in the AGC amplifier unit to measure the  $f_{11}$  element before active gain control with a separate low pass filter and amplification circuit. This data file, transferred directly to CPU memory then permanently stored on hard disk, is required information when converting between normalized and regular elements of the Mueller matrix. The following Figure 7 is a schematic drawing of the basic ellipsometer optical system and MCT detector with AGC circuit module. Figure AIII.11 (Appendix III) is a functional block diagram of an AGC amplifier circuit built for this system by Analog Modules, Inc.

In Figure 7, connections A and B are resonance frequencies driving oscillating birefringence in both transmitter and receiver PEM crystals: they are the phase reference points of the retardations along the ZnSe extraordinary axis. Connection D contains (normalized) phase information in the scattered beam radiance after AGC operation (the ac Mueller matrix components ratioed by the  $f_{11}$  element), and C contains the absolute magnitude of the scattergram signal before loop control (i.e., the dc component without AGC operation). The analog data acquisition system compares the phase of D to the reference phases A and B plus its respective combination frequency components (see Section 4.5.2).

The AGC amplifier must not introduce any propagation delay or phase shift of its own as signal strength varies. A beam chopper operating at a rate of  $\approx 100$  Hz is used to produce a dark time so that the detector offset voltage can be eliminated via a closed loop. The amplitude of the MCT signal will vary from  $1 \mu V$  to  $1 mV$ , due to changes in beam wavelength and backscatter angle experimental variables. The ac signals filtered by the lock-in circuit boards are between frequencies 1 and 100 KHz. The Analog Modules automatic gain control amplifier (Figure AIII.11), has the following salient characteristics: (1) gain control from 60-100 dB, 1% or better regulation from 1 KHz to 200 KHz, noise  $< 25 \mu V$ , frequency bandwidth 1 MHz to  $< 1$  KHz, an input impedance of  $100 K\Omega$ , and a  $\pm 15$  VDC supply voltage.

Module VII is an incident beam power regulation circuit. The split incident beam pyroelectric detector output is monitored by the power regulation circuit, and compares this value to a preset desired power reference. Depending on the comparison of these values, the servo motor will rotate the axis of a linear polarizer through which the incident beam transmits. Its function is to regulate the intensity of the beam incident to the scattering sample when switching between probe beams (course adjustment) and during irradiation (fine regulation). The circuits of this module are given in Figures AIII.12a-b.

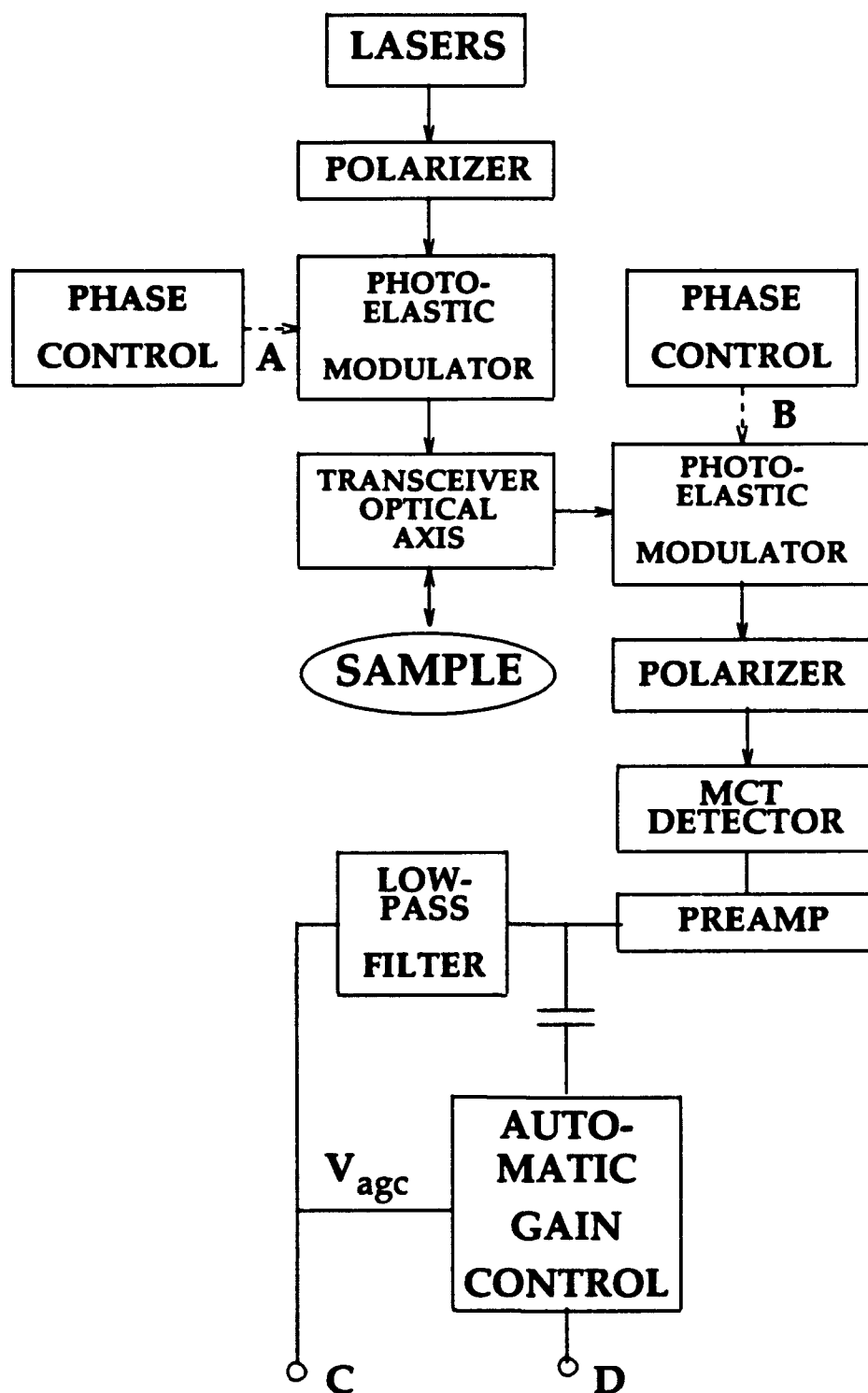


Figure 7. The ellipsometer's optical system, detector, and gain-control modules.

#### 4.5.5 Stepper Motor Control of Optics Hardware, Switching Between Incident Beams, PEM Peak Phase Retardation Selection Per Beam.

We call Module VIII the Serial Addressable Gateway (SAG) system of the analog APSD (Figure AIII.13a). This module is a communications bus consisting of four RS-232C serial ports and sixteen discrete I/O points. The SAG unit provides serial communications to each of the stepper controllers that automate the experimental operation, and to the A/D board that digitizes all data output channels. It also provides for eight control points used for shutter control for switching between beams (Figure AIII.15a), and for maintaining constant 2.404 rad peak modulator retardation in transmitter and receiver PEM's (Figure AIII.14) between switched beams of unlike energies. This module allows bi-directional communications between the host computer and associated devices it wishes to communicate with.

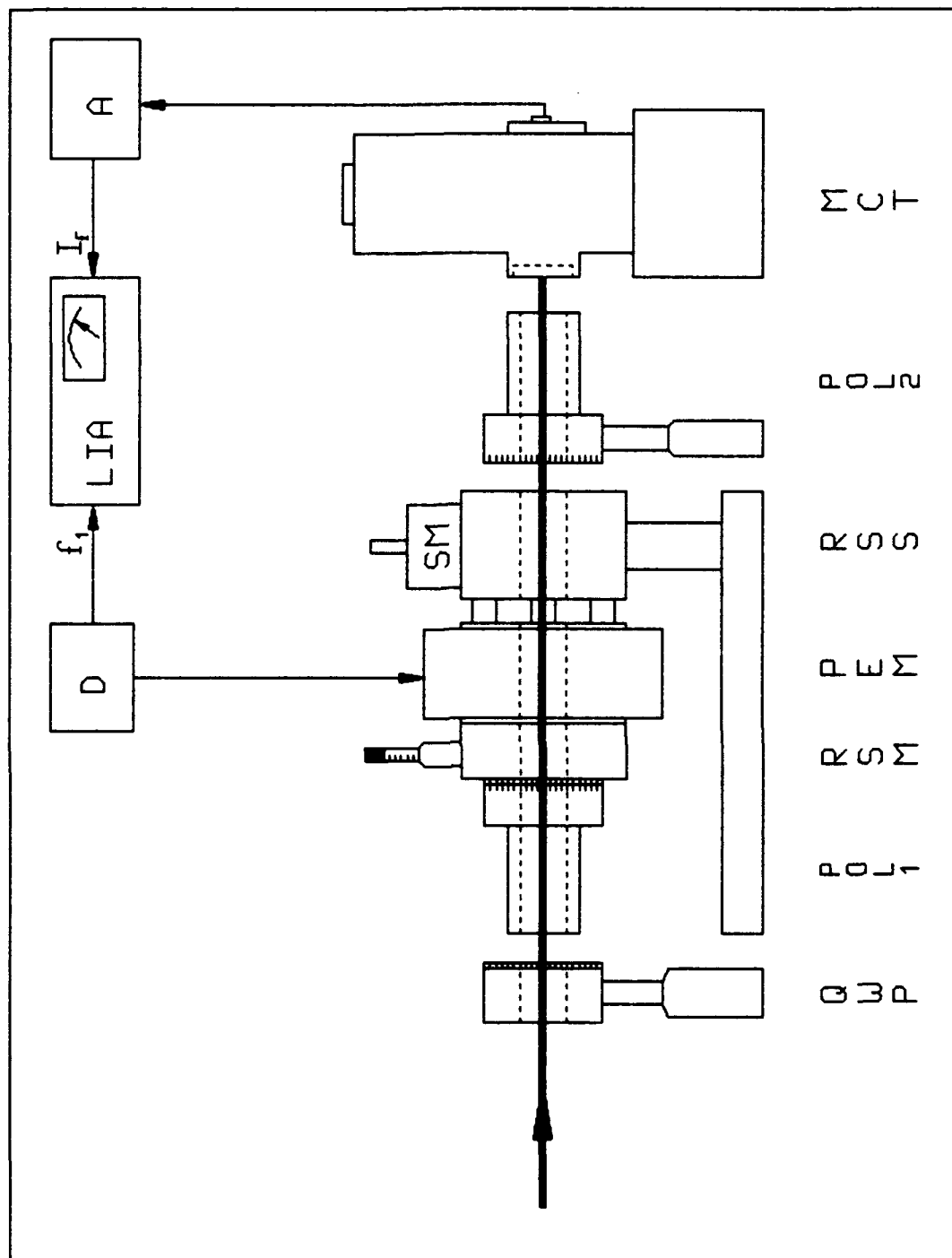
#### 4.6 Alignment and Calibration.

Topics discussed in this section are optic and electronic alignment methods for correct matrix file production and collection, and calibration procedures that transform measured matrix elements collected by the ellipsometer system (this applies only to the goniometer-based and field ellipsometers of Figures 4a and 4c) to the true surface-analyte sample Mueller elements.

##### 4.6.1 Alignment of Coupled Polarization Modulator and Linear Polarizer Axes.

Angle between the PEM's ZnSe principal (extraordinary) axis and mated linear polarizer transmission axis must be precisely  $45^\circ$  to insure pure polarization-modulation in irradiation beam and scattered radiance, thus maximum signal-to-noise ratios in the Fourier intensities (primary and overtone modulator frequencies) of the scattergram  $I_f$ . The calibration experiment for accomplishing precision alignment in PEM-polarizer axes is shown in Figure 8.

The quarter-wave plate optic QWP, Figure 8, converts the incident linearly polarized beam to circular polarization. Consequently, rotation of POL1 does not change beam intensity transmitted through the PEM-POL unit when rotated to produce cases A-D. Each PEM-POL unit consists of an IR linear polarizer (stacked Ge plates oriented at the Brewster angle) attached to a micrometer-adjusted rotary stage RSM attached to a photoelastic modulator PEM attached to a stepper-motor controlled rotary stage RSS. Linear polarizer POL2 produces, in conjunction with the active POL-PEM unit, intensity modulation in the beam striking the MCT photoconductive chip at the PEM transducer driving frequency (sine wave generator). This modulated output is pre-amplified through an ac-coupled circuit (A) and sent to the input channel of a lock-in amplifier (LIA). Reference frequency  $f_1$ , split from the transducer oscillator driving the ZnSe crystal to resonant vibration (located in the PEM head), is sent to the LIA's reference channel. The LIA electronically multiplies reference and detector waveforms, viz, it produces an analog output that tracks the cosine of the phase difference between reference and MCT sinusoids (bounded by  $\pm 5V$ ).



**Figure 8.** Alignment of the ellipsometer's linear polarizer (POL1) - photoelastic modulator (PEM) pairs. LIA, lock-in amplifier; D, electronic oscillation circuit driving the PEM at frequency  $f_1$ ; A, MCT detector amplifier; QWP, quarter-wave plate; RSM, precision rotary stage coupling POL1 and PEM; SM, stepper motor; RSS, stage for rotating the POL1-PEM pair; and MCT, liquid nitrogen cooled HgCdTe infrared photoconductive detector.

When aligning the optics between POL1 and PEM axes, the micrometer on RSM is turned until a null output is displayed on the LIA meter: the PEM retarder and POL1 transmission axes are now co-aligned. Two nulls will occur per  $2\pi$  POL1 revolution. It is good practice to rotate POL1 several times to insure that these nulls appear exactly  $180^\circ$  apart. An alignment precision of about 5 arc minutes between nulls is possible with this particular design.

Now that the polarizer transmission and modulator fast axes are co-aligned, rotation of POL1 (course then fine turning of RSM) exactly  $45^\circ$  puts the PEM-POL units into final alignment. A precision 5 arc minutes or better is possible, given the precision rotary stage RSM employed here. This alignment can also be performed more directly by rotating POL1 until a maximum output is displayed by the LIA. However, we find that locating nulls in the LIA's analog meter is a more accurate measurement technique, compared to seeking a maximum deflection during rotation of POL1. (Alignment error goes as the cosine of the offset angle between PEM-POL axes.)

#### **4.6.2 Amplitude and Phase Adjustments of the Detector's Nine Element Channels to Transmission Optics of Known Mueller Matrix.**

Calibration of the ellipsometer instruments can be performed routinely before and after experimental trials through measurements of spectral intensities in the scattergram (Equations 12, 14b, 16b, and 18b) by the  $\text{CO}_2$  beam transmitting three optic calibrators: (a) linear polarizer, (b) quarter waveplate, and (c) combination polarizer-waveplate and waveplate-polarizer; inserted between transmitter and receiver POL-PEM units. The measured Mueller matrix elements of the calibrator (Figure 9) are matched to its known elements by proper phase and gain adjustment of each of nine PSD boards designated Module IV of the analog acquisition system. (The phase per channel of the 15 ac elements are adjusted to match the known calibrator elements, and gain per channel with VGC activation is set to  $\pm 1$ . See Table 3b, and Section 4.5.) With VGC operation, the  $f_{11}$ -element channel is, of course, maintained to a preset voltage within the linear operating range of the MCT detector. The calibration experiment is summarized in Figure 9.

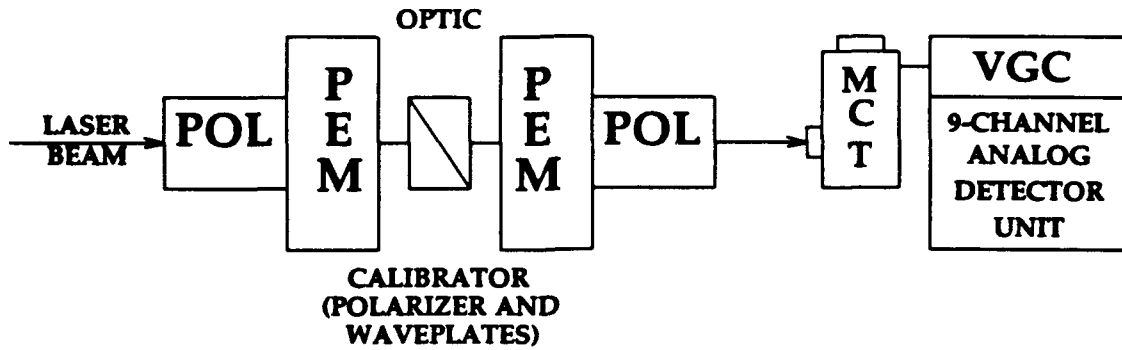


Figure 9. Calibration of the analog data acquisition channels for measurement of all Mueller matrix elements. PEM-POL are the transmitter and receiver photoelastic modulator-polarizer pairs, MCT is the infrared HgCdTe photoconductive detector, and VGC is its variable gain control amplifier (see Figure AIII.11). The calibrators are polarizer and waveplate optics of known Mueller matrix elements. Each channel of the analog detection unit has independent adjustments for phase and amplitude to match the calibrator signatures over the dynamic range of the MCT output waveform (see Section 4.5).

The calibrator optics exhibit Mueller elements of the form:

*Linear Polarizer*

$$P(\theta) = \frac{1}{2} \begin{bmatrix} 1 & \cos(2\theta) & \sin(2\theta) & 0 \\ \cos(2\theta) & \cos^2(2\theta) & \cos(2\theta)\sin(2\theta) & 0 \\ \sin(2\theta) & \cos(2\theta)\sin(2\theta) & \sin^2(2\theta) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (19a)$$

*Quarter-Wave plate*

$$Q(\rho) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos^2(2\rho) & \cos(2\rho)\sin(2\rho) & -\sin(2\rho) \\ 0 & \cos(2\rho)\sin(2\rho) & \sin^2(2\rho) & \cos(2\rho) \\ 0 & \sin(2\rho) & -\cos(2\rho) & 0 \end{bmatrix} \quad (19b)$$

where  $\theta$  is the transmission axis of the polarizer and  $\rho$  is the fast axis of the waveplate (quarter-wave retardation).

The rotating polarizer (Equation 19a) calibrates the Mueller elements;  $f_{31}$ ,  $f_{12}$ ,  $f_{21}$ , and  $f_{13}$ . Rotating quarter-wave plate (Equation 19b) calibrates  $f_{22}$ ,  $f_{23}$ ,  $f_{24}$ ,  $f_{32}$ ,  $f_{33}$ ,  $f_{34}$ ,  $f_{42}$ , and  $f_{43}$ . A combination of polarizer and quarter-wave optics in operator order  $P(0)Q(\rho)$  calibrates element  $f_{41}$ , while  $f_{14}$  can be calibrated in operator order  $Q(\rho)P(0)$ . A measurement with no optic (air) calibrates the channels for elements  $f_{11}$  and  $f_{44}$ .

The following Table 4, summarizes how the dc and eight lock-in frequency channels of the analog detection system (Section 4.4) are calibrated to the known rotating optic(s). The calibration of the detector can easily be checked before and after running the ellipsometer system for lengthy periods.

Table 4. Calibration of the PSD's analog electronic channels to the Mueller elements of a rotating quarter-wave plate  $Q(\rho)$ , rotating linear polarizer  $P(\theta)$ , and combination optics  $Q(\rho)P(0)$  and  $P(0)Q(\rho)$ .

Channel	Lock-in Frequency	Incident, Final Stokes Vectors [Calibrator, Mueller Element, Signal]			
		V, V	V, -45°	+45°, -45°	+45°, V
1	dc	[Air, 11, unity]			
2	$\omega_2$	[ $P(0)Q(\rho)$ , 41, $\frac{1}{2} \sin 2\rho$ ]			
3	$\omega_1$	[ $Q(\rho)P(0)$ , 14, $-\frac{1}{2} \sin 2\rho$ ]			
4	$2\omega_2$			[ $P(\theta)$ , 31, $\frac{1}{2} \sin 2\theta$ ]	[ $P(\theta)$ , 21, $\frac{1}{2} \cos 2\theta$ ]
5	$2\omega_1$		[ $P(\theta)$ , 12, $\frac{1}{2} \cos 2\theta$ ]	[ $P(\theta)$ , 13, $\frac{1}{2} \sin 2\theta$ ]	
6	$\omega_1 + \omega_2$	[Air, 44, unity]			
7	$2\omega_2 + \omega_1$	[ $Q(\rho)$ , 24, $-\sin 2\rho$ ]	[ $Q(\rho)$ , 34, $\cos 2\rho$ ]		
8	$2\omega_1 + \omega_2$		[ $Q(\rho)$ , 42, $\sin 2\rho$ ]	[ $Q(\rho)$ , 43, $-\cos 2\rho$ ]	
9	$2\omega_1 - 2\omega_2$	[ $Q(\rho)$ , 22, $\cos^2 2\rho$ ]	[ $Q(\rho)$ , 32, $(\cos 2\rho)(\sin 2\rho)$ ]	[ $Q(\rho)$ , 33, $\sin^2 2\rho$ ]	[ $Q(\rho)$ , 23, $(\cos 2\rho)(\sin 2\rho)$ ]

#### 4.6.3 Decoupling Sample From System Matrix Elements in the 3-Mirror Goniometer Type Ellipsometer Waveform Output.

Although any discrimination between bare and contaminated surfaces with this instrument will rely principally on the wavelength dependence of the backscattered light, we wish also to investigate the effect of varying the angle at which the IR beam strikes the sample surface. The simplest means for accomplishing this would be to fix and direct the incident beam straight downward onto a sample holder (e.g., a petri dish) and tilt the holder through the desired range of angles with an appropriate mount. Unfortunately we will look at numerous loosely packed samples, such as soils, coated with liquid contaminants, so only a small tilt angle would be allowed. Instead, then, we have chosen to lay our porous granular samples flat on the optical table and vary the light's incident angle using a goniometer - whose mirrors also return the backscattered radiation. By not disturbing the scatterer, the natural effects of liquid diffusion (into and across the porous material bulk) and evaporation of the analyte can be analysed by a screening of elements sensitive to surface geometry.

The drawback to this arrangement is that the instrument measures the net Mueller matrix of everything in the optical path between the two polarization modulators, so the sample's Mueller matrix is buried in the middle of a long product of matrices representing all the goniometer mirrors - both going in and returning (Equation 6b).

We now return to the real-time scattergram obtained from the ellipsometer configuration of Figure 4a, the configuration with the goniometer transceiver arm. This complex waveform requires filtering of the arm's mirror elements for extraction of the Mueller elements of the scattering sample.

In Equation 6b,  $M_i$  represent four metallic mirror matrices oriented in a plane with normal vector  $45^\circ$  to the incident beam. Three of these mirrors make up the goniometer arm, and the other directs backscattered radiance from sample to MCT detector. The mirror optics and arm rotation matrices are given in Table 2. The unknown sample Mueller matrix is embedded in the system matrix  $\psi_{ij}$  measurements, and needs to be extracted. This can be done analytically by simply inverting M and R from the left and right hand sides of Equation 6b, and substituting element values from Table 2, given values of refractive indices supplied by the manufacturer of the optical surfaces. Each mirror Mueller matrix is an exact function of the optical 'constants' n and k (in Table 2,  $\alpha$ ,  $\beta$ , and  $\sigma$  are implicit functions of n and k) of its surface coating layers and substrate material. Even under strict quality control procedures from the manufacturer, all mirrors cannot be assumed to have identical  $\sigma$ ,  $\beta$ , and  $\sigma$  values. Since their variance in n and k values are not accurately known, three separate experiments are required for empirical determinations of  $M_1M_2M_3$ ,  $M_3M_2M_1$ , and  $M_3$ .

The calibration experiments for decoupling the mirror Mueller elements from the detected signal are schematized in Figure 10. In the top configuration, we define:  $B(\lambda) = M_3(\lambda)M_2(\lambda)M_1(\lambda)$ ; according to the order in which the incident beam reflects from the mirror flats and transmits through receiver the optics (R) to the MCT detector. The  $\Gamma$  and R optics are interchanged, and beam direction reversed in the middle configuration of Figure 10, yielding the inverse-order goniometer matrix we define:  $C(\lambda) = M_1(\lambda)M_2(\lambda)M_3(\lambda)$ . In the final calibration measurement for  $M_4$ , bottom POL-MOD configurations of Figure 10,  $D(\lambda) = M_4(\lambda)C(\lambda)$ . Note, in the above cases, the goniometer is fixed in the plane of incidence, i.e.,  $\theta = 0^\circ$  and  $R = I$  - the identity matrix. The matrices  $M_i$  are noncommutative, i.e.,  $B(\lambda) \neq C(\lambda)$ . By inverting Equation 6b, and substituting in the calibration matrix data, we come to a solution for the Mueller matrix of the contaminated sample with the following form.

$$F(\lambda, \theta) = C^{-1}(\lambda)R^{-1}(\theta)D^{-1}(\lambda)C(\lambda)\Psi(\lambda, \theta)R^{-1}(\theta)B^{-1}(\lambda) \quad (20)$$

Presented in Appendix II are term-by-term values the above F product matrix. The result was symbolically computed from a LISP coded mathematical algorithm named MACSYMA, and were later checked for accuracy. Fortran 77 code of the element equations, also produced by MACSYMA, is used in the decoupling software conversion operation labeled  $\psi_{ij} \rightarrow f_{ij}$  in Figure 6.

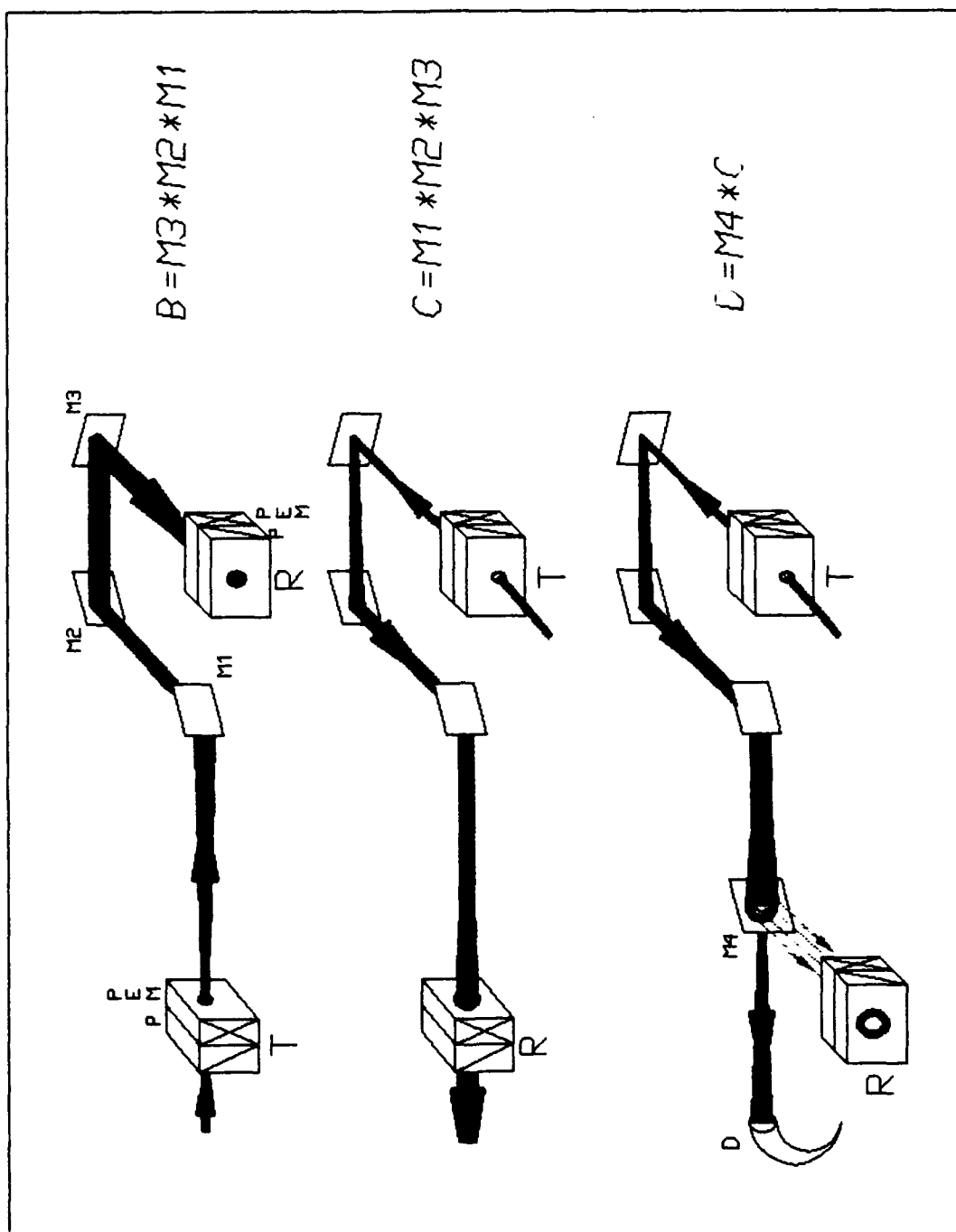


Figure 10. Calibration experiments for decoupling four mirror (M1-4) matrix elements from the ellipsometer system matrix of Figure 4a. T and R are the transmitter and receiver linear polarizer(P) - photoelastic modulator(PEM) pairs, respectively, and D is a beam dump. The goniometer arm is oriented + or  $-90^0$ , so that the reference plane of measurement of the Stokes vectors in transmitted and received beams are the same. Mueller matrices B, C, and D are produced from the respective optical orientations, and must be measured for each laser wavelength.

### Optical Redesign of the 3-Mirror Goniometer Arm

In principle, the Mueller sample matrix can be extracted from the measured system matrix if the matrices of the mirrors are known. Sections 4.2 and 4.6.3 details the calibration measurements needed - at every wavelength! - for a Mueller description of the goniometer and the subsequent calculations to deconvolve the desired sample matrix from the total measured system matrix. This process clearly is not satisfactory. It is at best time consuming and inelegant, and, more seriously, there are unresolved questions about the propagation of uncertainties in the goniometer calibration measurements into the calculation of the final matrix. We consider it more sound for the goniometer arm to be redesigned so that the mirror matrices are measured empirically with the instrument rather than to rely on theoretical calculations that, though precise in form, require an exact knowledge of the mirror surfaces' IR optical constants.

Lets now return to the Mueller matrix,  $M$ , of a mirror previously expressed in Table 2.

$$M = \frac{1}{2} \begin{pmatrix} \alpha^2 + \beta^2 & \alpha^2 - \beta^2 & 0 & 0 \\ \alpha^2 - \beta^2 & \alpha^2 + \beta^2 & 0 & 0 \\ 0 & 0 & -2\alpha\beta\cos\sigma & 2\alpha\beta\sin\sigma \\ 0 & 0 & 2\alpha\beta\sin\sigma & -2\alpha\beta\cos\sigma \end{pmatrix} \quad (21)$$

As before, the Stokes vectors of the incident and reflected rays are both referred to the plane of incidence containing those two rays (and the mirror normal),  $\alpha$  is the ratio of reflected to incident amplitudes for light polarized parallel to the plane of incidence,  $\beta$  is the same ratio for light polarized perpendicular to the plane of incidence, and  $\sigma$  is the reflection induced phase shift between the two components. Note that there is no mixing between parallel and perpendicular polarization components.

Figure 11a illustrates the simple case of a mirror reflecting a light beam upward by  $90^\circ$ . Let the incident beam, segment 1, have a Stokes vector  $(s_0, s_1, s_2, s_3)$  originally referred to the horizontal plane, H. To apply Equation 5, the incident Stokes vector must first be re-referenced to the reflection plane of incidence, V. This is done by operating with the rotation matrix  $R$ , defined as:

$$R(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 2\theta & \sin 2\theta & 0 \\ 0 & -\sin 2\theta & \cos 2\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (22)$$

where  $\theta$  is the angle between the old and new reference planes.

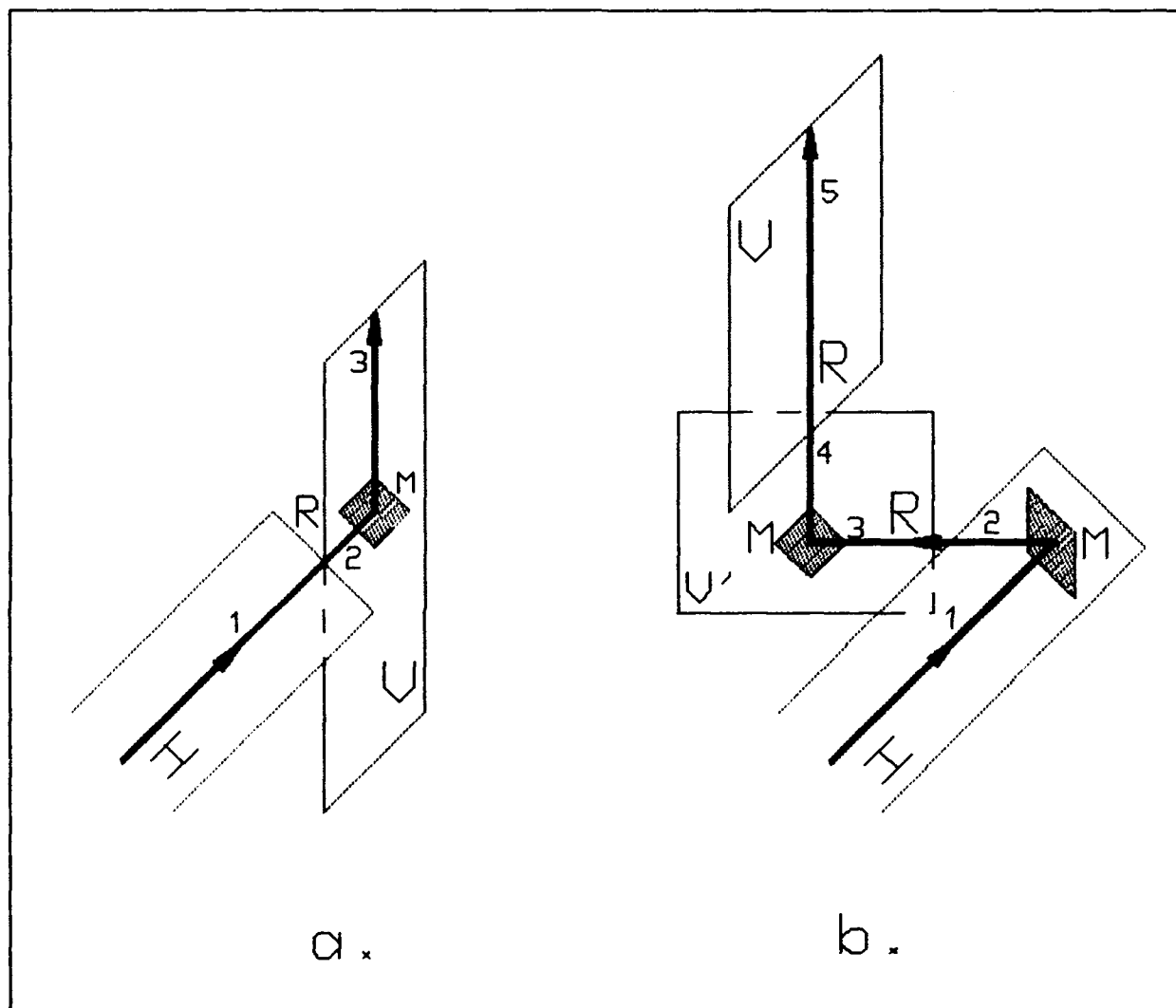


Figure 11. A right angle reflector (a) with a single mirror that changes the Stokes vector and (b) with a pair of mirrors that do not.

Thus, the Stokes vector of the reflected beam, segment 3, referenced to plane V is:

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix}_3 = M R(90^\circ) \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix}_1 \quad (23)$$

Notice that  $M R(90^\circ)$  is just  $M$  with sign changes in the 2<sup>nd</sup> and 3<sup>rd</sup> columns.

In Figure 11b we also direct an initially horizontal beam upward, but this time it is first reflected  $90^\circ$  in the horizontal plane. A final application of  $R(90^\circ)$  between segments 4 and 5 makes the final beam (segment 5) identical with respect to direction and reference frame as the final beam (segment 3) in Figure 11a, but now

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix}_5 = R(90^\circ) M R(90^\circ) M \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix}_1 \quad (24)$$

Performing the suggested matrix multiplications, we easily find

$$R(90^\circ) M R(90^\circ) M = \alpha^2 \beta^2 \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (25)$$

The Stokes vector of the final beam is identical to that of the initial beam, except for an unimportant attenuating factor  $\alpha^2 \beta^2$ . A final reference rotation is not critical and was applied to make Figures 11a and 11b exactly comparable. The only difference without it is a change in two signs of the unity matrix.

The reason a pair of mirrors arranged as in Figure 11b is transparent with respect to Mueller calculations is, of course, that the identity of parallel and perpendicular polarization components is interchanged for the two reflections. The second mirror reverses the relative phase shift of the first mirror and also equalizes the amplitude attenuations. Also note that this result holds for every wavelength. We require only  $90^\circ$  reflections and that the mirrors be optically identical.

We believe a goniometer can be constructed on this principle and will allow a direct probing of the sample's Mueller matrix without all the calibrations and inverse matrix calculations of Section 4.6.3.

#### **4.7 System Software: Experiment Automation, Data Collection and Graphics Display.**

The development of software for system hardware automation, analog-to-digital conversions for data collection, and graphics presentation for visualization of reduced data sets is updated as new experiments are devised. The first version ellipsometer software package is complete, and is presented in Appendix IV. Linked to the hardware of the analog data acquisition system, the design of this automation and data graphics analysis code is modular. Structured in menu format, it is flexible enough to incorporate changes for accommodating future applications of these ellipsometer systems. A VAXstation II/GPX computer operating under VMS version 5.4 controls automation I/O between it, the SAG, and the DAEDAL MC2000 series stepper motor controllers. All system software is written in the FORTRAN 77 language.

##### **4.7.1 Switching of Laser Shutters, Modulator Retardation Adjustment.**

The laser shutters and the modulator retardations are controlled from the 'NEW\_TEK.FOR' (Appendix IV) routine as string commands to the main controller relay SAG network (see Appendix III, Figures AIII.13a-b, 15b).

##### **4.7.2 Modulator-Polarizer Permutations.**

These are stepper motor controller functions. Movements are predetermined by the user and stored on file. Refer to 'MOV\_STAGE.FOR' and associated routines (Appendix IV).

##### **4.7.3 Sample Selection and Stage Rotation.**

Sample selection is based on the input order in which a series of dry and wetted surface measurements are made, with a maximum of eight samples. The means by which a sample is selected and rotated about its axis is a function of the 'NEW\_TEK.FOR, VV.FOR, V45.FOR, P45V.FOR, and P4545.FOR' routines (Appendix IV). The latter four routines, named for their associated POL-PEM axis orientations, index the sample stage rotation so that Mueller elements can be collected at any range of backscattering angles to and from the various dry and wetted sample scatterers.

##### **4.7.4 Goniometer Rotation and Data Acquisition.**

Goniometer rotation operations is performed by 'VV.FOR, V45.FOR, P45V.FOR, and P4545.FOR' routines (Appendix IV). The goniometer is controlled in an identical manner as in the sample stage rotation routines, with the exception that its angle increment can be adjusted to provide measurements at any resolution. Data acquisition is contained in these four subroutines and occurs sequentially and synchronous to stage rotation.

##### **4.7.5 Data Storage and File Management.**

The collected sample data is stored in two discrete files. One is an index file that contains information about the sample, the other file contains the Mueller matrix data. Refer to the 'NEW\_TEK.FOR' routine (Appendix IV).

#### **4.7.6 Graphics Presentation.**

The graphics display for this program was written for a 4111 TEKTRONIX color terminal or equivalent. (Graphics are not required to run the experiment.) The graphics routines that display Mueller elements of the scatterer as functions of backscattering angle and beam wavelength are 'TEK\_INPUTS.FOR, TEK\_TEXT.FOR and LASER\_IN.FOR' (Appendix IV). For real time graphics display refer to 'TEK3.FOR, DRAW\_ELE.FOR AND SEE\_ELE.FOR' (Appendix IV).

The following Figure 12 is a typical graphics output of the software package given in Appendix IV.

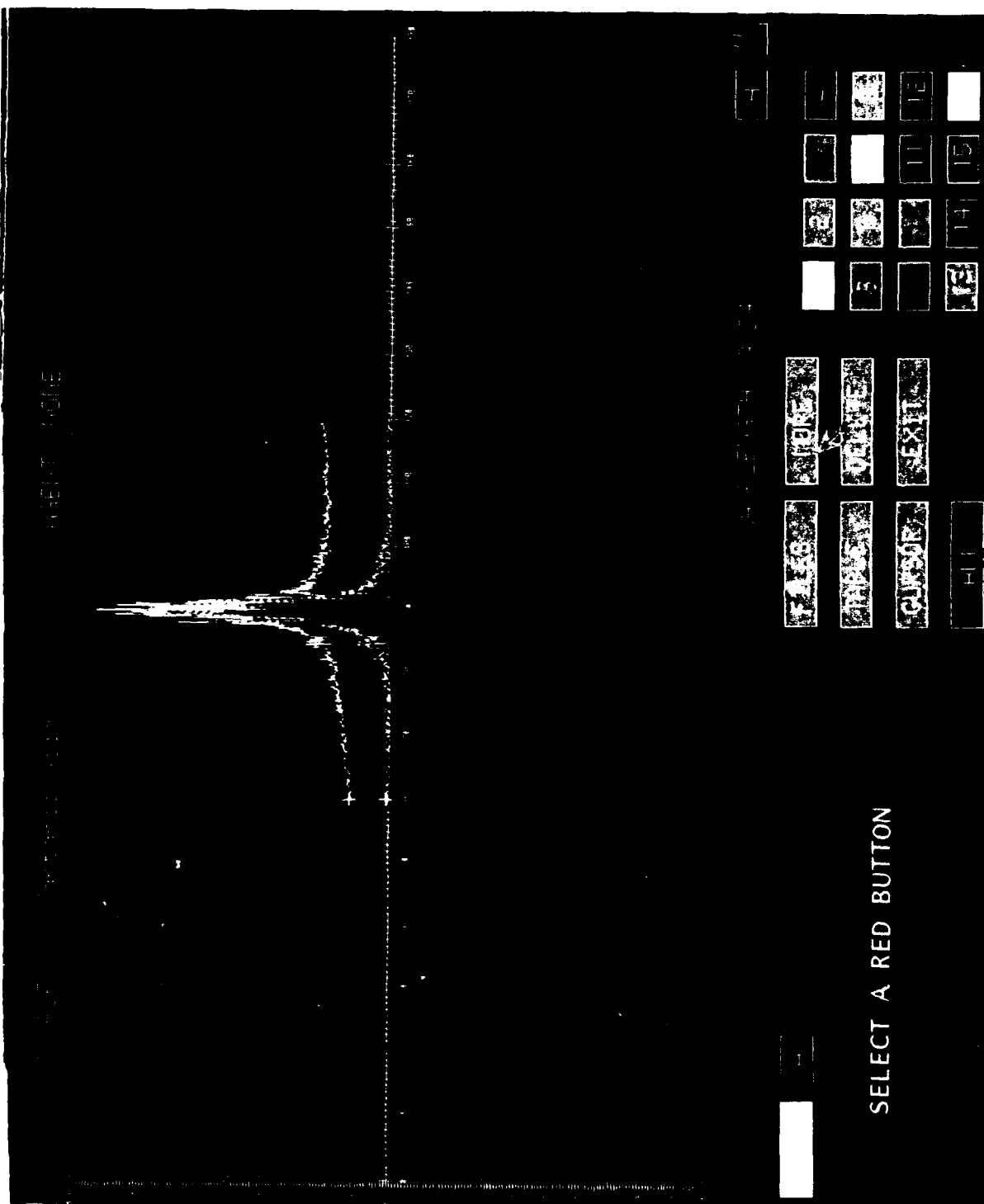


Figure 12 Graphics output from the software package of Appendix IV. In this particular data set, 12 measured elements are displayed as a function of backscattering angle ( $0.1^\circ$  resolution) from a wafer of isomer (-) tartaric acid at  $\lambda = 9.24 \mu\text{m}$ . The color-coded Mueller field of elements is displayed in the the lower right. Boxes H and V are for horizontal and vertical axes scrolling, box CURSOR activates H and V, box I is for the selection of any of the Mueller elements per PFM optics orientation, box FILES contains the data files, box TABLE contains the raw voltages output from the A/D channels, box DELETE erases element(s) from the screen, box ALL with DELETE erases all elements from the screen, box MORE selects additional elements for display.

## 5. THEORETICAL MODELLING

Modelling of the Mueller matrix elements is being performed using: (1) RETRO/DISPLAY, a Full Wave theoretical software package that provides closed form expressions for scattering of electromagnetic waves from isotropic surfaces of 'rough' through 'smooth' texture, and graphical analysis of those data; (2) DETECT/DECIDE2, algorithms that select optimum wavelengths and angles from RETRO for maximum probability of contamination detection and; (3) CADPAC/BROOKLYN89/GAUSSIAN88, quantum chemistry software packages that predict infrared, equilibrium geometries, vibrational modes and frequencies, vibrational circular dichroism and other physical properties of the target molecules.

Currently, one of us (J.O. Jensen) along with researchers at the University of Pennsylvania (H. Hamelka), Lehigh University (D. Zeroka), the Ballistics Research Laboratory (C. Chabalowski), are modifying the CADPAC/BROOKLYN89/GAUSSIAN88 packages for infrared absorption and VCD spectral interpretations inherent in Mueller elements [1,4], [4,1], and [1,1]. (The elements  $f_{14}$  and  $f_{41}$  contain information on the VCD property. BROOKLYN89 is now being modified for VCD calculations by various chiral molecules. Also, CADPAC/GAUSSIAN88 is now being used by D. Zeroka for vibrational modes and VCD predictions of linear and ringed sugars and amine molecules that simulate chirality in the more complex biological structures.) There is an interest here to build on a valid quantum chemistry model predicting IR absorption in the more complex molecular systems that behave like chemical/biological agent compounds, and couple it to a tested Full Wave scattering theory. (A purely analytical model of scattering. The present Full Wave model must access physical information on the scatterer from an experimental data bank.) This model's output would in turn transfer it output to a neural network connected to the data channels of the ellipsometer sensor (Section 6.3).

We give a brief summary on quantum modelling approaches in the following section. A more detailed discussion of vibration-rotation, VCD, depolarization and other properties of the analyte compounds of interest will be presented in future papers. Moreover, we do elaborate on and give the source code of model RETRO/DISPLAY and its associated algorithm DETECT/DECIDE2 later in this section and Appendixes V and VI.

### 5.1 Modelling the Analyte's Resonant Molecular Motions: Applying CADPAC, BROOKLYN89, and GAUSSIAN88 Quantum Chemistry Codes.

The backscattering of polarized light to yield a Mueller matrix depends, in part, on surface geometry. This has been addressed by the work of E. Bahar et al (Sections 5.2.1 - 5.2.4, and Appendix V). Another (coupled) aspect of the Mueller matrix is the interaction of light with specific molecules on the surface, a part associated with the pure physical nature of the scatterer. The molecular phenomena causing these interactions include absorption, depolarization, and circular dichroism. These molecular phenomena will result in an index of refraction that is in a matrix form similar to the Mueller matrix itself. Thus the index of refraction and absorption coefficient are no longer simple scalar quantities. If this matrix form of the refractive index can be extracted from the data, useful chemical data can then be determined.

From the quantum chemistry codes GAUSSIAN90<sup>25</sup>, CADPAC<sup>26</sup>, and BROOKLYN89<sup>27</sup> we can accurately predict absorption, depolarization, and circular dichroism in the gas phase of molecules. The spectra of a molecule on a surface is similar to a gas phase molecule. Thus we can accurately predict the matrix form of the refractive index by the interaction of light with the given molecule.

Vibrational Circular Dichroism (VCD) is particularly useful in predicting biological contaminants.<sup>28</sup> It is a measure of scattering between left- and right-handed circularly polarized light, interactions that differ with chiral molecules. VCD is related to the [4,1] and [1,4] elements of the Mueller matrix, since these elements are transformations of one circular handedness into another. If the [4,1] and [1,4] elements are measured at two different frequencies on and off resonance, then the difference between the scattering intensity in these elements is indicative of the presence or absence of a chiral molecule.

The flowchart of the calculations that we perform is given in Figure 13. The calculations are usually done at the Hartree-Fock level of theory using a finite basis of Gaussian type wave functions. The minimum energy configuration of the analyte is found within this approximation. The second derivative with respect to all nuclear displacements is then found. The eigenvalues and eigenvectors of the second derivative matrix give the vibrational frequencies and normal modes, respectively. The dipole derivatives along the normal modes give the relative intensities of the peaks.

VCD is calculated using the computer package CADPAC. VCD is a non-Born-Oppenheimer effect caused by the coupling of electronic and nuclear motion. The calculation gives the overlap of the change in the wave function due to a nuclear displacement against the change in the wavefunction due to an external magnetic field. Thus the nuclear motion causes a slight asymmetry in the way that left-handed and right-handed circularly polarized light interact with the molecule.

It is known from the literature<sup>29</sup> that vibrational frequencies calculated at the Hartree-Fock level of theory tend to be slightly higher than experiment. The correction factors needed to make the calculated frequencies agree with the experimental frequencies tend to be constant across a group of similar compounds. Much of the work done in our laboratory involves determining the correction factors to the raw calculated results.

## OVERVIEW OF THE SPECTRAL COMPUTATION PROCEDURE

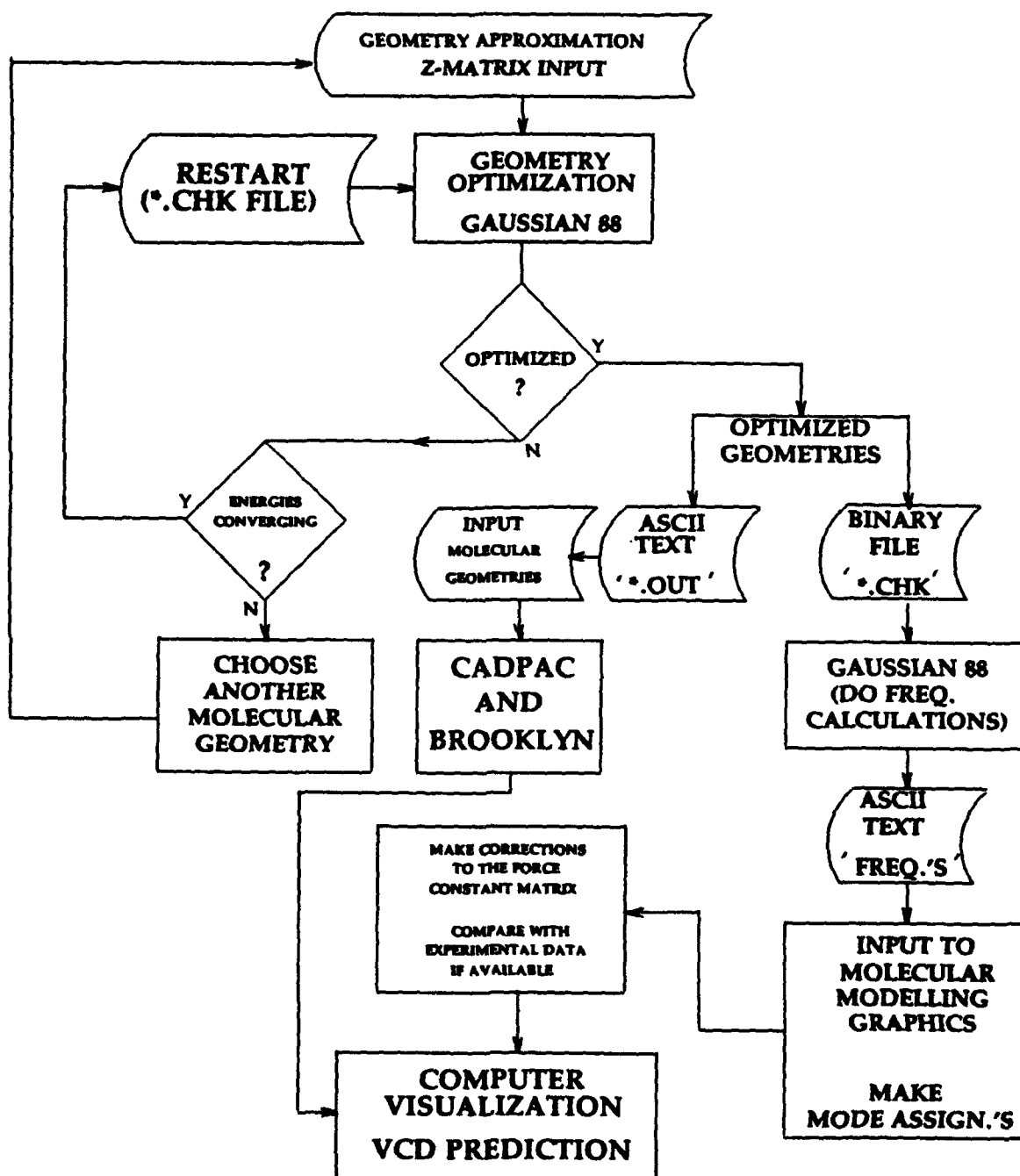


Figure 13. An outline of the computation procedure used to predict the analyte's IR spectrum via Gaussian, CADPAC, and BROOKLYN quantum chemistry software packages.

### **Linkage to the Full Wave Scattering Code.**

In the latter Section 6.3, we discuss a concept in which the ellipsometer's real-time Mueller element outputs are connected to the inputs of a neural network. This network would be designed to partition analyte-specific data, so that presence or absence of the analyte can be established in the scattering zone of the ellipsometer probe beams, as indicated by an appropriate alarm signal at the network's output layer. Inputs to the neural network are weighed according to a valid theoretical model. A candidate model for scattering by a randomly rough surface is the Full Wave theory presented in the following section. We anticipate making a linkage of the CADPAC, GAUSSIAN, and BROOKLYN quantum chemistry codes to the Full Wave code. In a successful neural network model, the quantum chemistry codes would predict resonant absorption by the analyte(s). The scattering code accesses this information and outputs Mueller elements of the contaminated surface. That data would then be used to weight the network's input sensor data, and train it to alarm under certain conditions, i.e., when a susceptible set of Mueller elements are present and successfully partitioned from its background. An additional layer can be designed into the neural network that accesses a data bank where Mueller signal strengths and analyte mass densities are correlated, allowing a quantitative map of the threat contaminant to be displayed.

The quantum models consider single molecules, and are used to compute energies driving resonant molecular vibrational motion, assigning an intensity value to the absorption. In the Full Wave model, macroscopic boundaries separate material media and each material is characterized by a complex dielectric constant. Obviously, a macroscopic analyte medium containing many, many, molecules would likely have a broadening effect on the quantum predictions and, perhaps, can be predicted by the many-body theories of statistical mechanics.

Once the absorption of a material is determined by the quantum codes (over a wide spectral band), its real and imaginary parts of refractive index can be computed by the Kramers-Kronig<sup>10</sup> relationships. These are the data the Full Wave model uses in its surface scattering Mueller matrix predictions.

## **5.2 Mueller Matrix Predictions by EM Wave Scattering From Rough Surfaces: The Full Wave Model of Physical Optics Theory and its Application for Remote Detection.**

An experimental verifiable model for accurate predictions of the Mueller matrix elements is of great value in the development of a detection system. Predicting the Mueller elements as functions of laser beam scattering wavelength, statistical orientation between scatterer and incident beam, optical properties of the scatterer (indices of refraction), and its topography could essentially simulate the entire experimental operation. With these descriptive models, we seek an optimum domain for which these parameters can most readily reveal the analyte, and guide the experimenter toward a most probable detection scattering event.

### **5.2.1 The Full Wave Model: RETRO.**

Program RETRO is a numerical implementation of a Full Wave electromagnetic scattering theory developed by Professor Ezekiel Bahar at the University of Nebraska-Lincoln. This theory bridges the gap between physical optics and perturbation theories.<sup>30,31,32,33,34</sup> RETRO, written by Craig Herzinger, calculates theoretical Mueller matrix elements for light scattering by randomly rough 2-d surfaces. Associated software named DISPLAY presents 3-d, contour, and 2-d graphics from the output of RETRO. RETRO/DISPLAY was written specifically for this experimental program, and, should it be proven feasible, aid us in choosing optimum wavelength, angle, and polarization parameters for characterizing contaminants on various

background and interferent terrestrial or manufactured material. Later in this section we discuss what simplifying assumptions are made in Full Wave theory to allow numeric results to be calculated in a reasonable period of time. The RETRO/DISPLAY source code, written in Fortran 77 and now running on a CRAY supercomputer, is included Appendix V. Also, notational differences between the code and theory are addressed, as are format and content of input and output data files, and the relationships of many program variables to the theory in symbolic form.

Full Wave theory calculates scattering of an electromagnetic plane wave from a two dimensional, statistically rough, surface between free space (air) and a material with a relative dielectric constant,  $\epsilon_r(\lambda_0)$ , where  $\lambda_0$  is the free space wavelength of the radiation.

The surface boundary is defined as  $y = h(x, z)$ , where  $\langle h \rangle = 0$ .  
The reference plane is defined as  $y = 0$ .

The plane of incidence is assumed to be the x-y plane and  $\theta_0^i$  is the angle between the direction of the wave and the normal to the reference plane. For backscatter, the incident and final directions lie on the same line so that  $\theta_0^f = -\theta_0^i = \theta_0$ .

The mean squared height of the surface is  $\langle h^2 \rangle$ .

The mean squared slope of the surface is  $\sigma_h^2 = \langle h_x^2 + h_z^2 \rangle$ ,  $h_x = \frac{\partial h}{\partial x}$ ,  $h_z = \frac{\partial h}{\partial z}$ .

The correlation length of the surface,  $l_c$ , is defined by  $l_c^2 = \frac{4\langle h^2 \rangle}{\sigma_h^2}$ .

The auto-correlation function of the surface heights is  $r_{hh}(x_d, y_d) = \frac{\langle hh' \rangle}{\langle h^2 \rangle}$ ,

where  $h = h(x, z)$ ,  $h' = h(x', z')$ ,  $x_d = x - x'$ ,  $z_d = z - z'$ .

Full Wave theory allows calculations of elements  $F = F(\langle h^2 \rangle, \sigma_h^2, \lambda_0, \theta_0)$ .

The assumptions made in this program can be broken into two classes:

- Process assumptions and
- Surface assumptions.

#### *Process Assumptions:*

Process assumptions address how the light is scattered, for example how many times it strikes the surface, how the emitter and receiver are oriented, and how diffuse the scattered light is. Surface assumptions deal with the statistical representation of the surface heights and slopes.

This work assumes that scattering can be properly characterized by a single-scatter process. That is, light measured at the detector is assumed to have struck the rough surface exactly once; multiple scattering is not considered. This allows a second order iterative solution to be used in the Full Wave theory, additionally, it also limits how rough the surface can be. Another assumption is that the light source and detector are on the same optical path with the same orientation. The common terminology for this is backscattering. Backscattering is considered because the ellipsometer is a proving instrument for a future remote sensing system, where the receiver and detector are at one location far from the target surface.

The Mueller matrix elements are calculated on a per solid unit angle basis, therefore the matrix is correct to within a scalar constant of its experimental counterpart. The scalar constant is based on the size of the solid angle intercepted by the detector. Moreover, for this work to be valid, the detector must look at a small enough range of received radiation centered in the backscattering plane such that pure backscattering can be used as a good approximation of radiance collected from the irradiation cross-sectional area over the entire range of

angles and wavelengths. Also, the solid angle intercepted by the detector must be invariant to changes in incident angle,  $\theta_0$ , and wavelength,  $\lambda_0$ . A third assumption is that the scattered radiation is totally diffuse. This means the coherent portion of the return (the return without the surface roughness, times a constant) is assumed equal to zero. This condition limits how smooth the surface can be.

#### Surface Assumptions:

The first major assumption is that the surface is isotropic and uniform, i.e., the scattering is independent of the rotational or translational position of the rough surface; the scattering is invariant to a rotation or translation of the x and z axes. This leads to the conclusion that  $r_{hh}(x_d, z_d) = r_{hh}(r_d)$ ,  $r_d^2 = x_d^2 + z_d^2$ .

The second assumption is that the probability density of the heights and slopes are independent,  $p(h, h_x, h_z) = p_h(h)p_s(h_x, h_z)$ . Also,  $p_h$  and  $p_s$  are assumed, respectively, to be Gaussian and jointly Gaussian probability density functions.

To meet the condition of single scattering, the following restriction is applied:  $\sigma_s^2 \leq 2$ . To satisfy the condition of purely diffuse scattering,  $\langle h^2 \rangle$  is restricted by the relationship  $4\langle h^2 \rangle k_0^2$  is much greater than 1, where  $k_0 = 2\frac{\pi}{\lambda_0}$ . These conditions are not enforced by the program; you must make sure the input data is satisfactory.

#### Consequences of Assumptions:

Full Wave theory, under the above conditions/assumptions, can be used to calculate the scattering phase (Mueller) matrix  $F'$  as defined by the modified Stokes vector notation.

$$F' = \begin{pmatrix} \sigma_{VV}^{VV} & \sigma_{VH}^{VH} & 0 & 0 \\ \sigma_{HV}^{HV} & \sigma_{HH}^{HH} & 0 & 0 \\ 0 & 0 & \text{Re}\{\sigma_{VV}^{HH} + \sigma_{VH}^{HV}\} & \text{Im}\{\sigma_{VV}^{HH}\} \\ 0 & 0 & -\text{Im}\{\sigma_{VV}^{HH}\} & \text{Re}\{\sigma_{VV}^{HH} - \sigma_{VH}^{HV}\} \end{pmatrix} \quad (26)$$

$$\text{where } \sigma_{ij}^{kl} = Q \int_{-\infty}^{\infty} \frac{D^{ij} D^{kl*}}{(\bar{n} \cdot \vec{a}_y)^2} P_2 P_s dh_x dh_z \quad (27)$$

$$\text{and } Q = 2k_0^2 \int_0^{\infty} \left( \exp(v_y^2 \langle h^2 \rangle (1 - r_{hh})) - \exp(-v_y^2 \langle h^2 \rangle) \right) J_0(v_{xz} r_d) r_d dr_d \quad (28)$$

Note that under the process assumptions, eight elements of  $F'$  are analytically zero. Measuring these eight elements experimentally should be a good test of theory with the discussed restrictions.

#### Linkage to Scattering Matrix F':

In our previous notation,  $F'$  was a modified matrix derived from modified Stoke's vectors. To transform this matrix into the desired Mueller matrix  $F$ , the notational difference must be addressed. Consider a Stokes vector in the modified notation,  $I_M$ , and in the standard notation,  $I_S$ , that represent the same light.

$$I_M = \begin{pmatrix} S_0 \\ S_1 \\ s_3 \\ s_4 \end{pmatrix}, \quad I_S = \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} S_0 + S_1 \\ S_0 - S_1 \\ s_2 \\ s_3 \end{pmatrix} \quad (29)$$

The modified Stoke's vector of the scattered light is  $I_M' = F' I_M$  and the standard Stoke's vector of the scattered light  $I_S' = F I_S$  clearly must represent the same light. For this to be true

$$F = A F' B, \quad \text{where} \quad (30)$$

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (31)$$

$A$  and  $B$  are transformation matrices such that  $I_S = A I_M$  and  $I_M = B I_S$ .

RETRO then calculates  $F$  by computing  $Q$  and the necessary  $(\sigma_{ij}^k / Q)$ 's.

#### 5.2.2 DISPLAY: Graphical Analysis of the Full Wave Model.

Program DISPLAY is an interactive program that plots elements of backscatter Mueller matrices. The elements can be displayed in a 3-d, 2-d, or contour format as a function of radiation wavelength,  $\lambda_0$ , and incident angle,  $\theta_0$ . The program is written in FORTRAN 77 and is set up to run on the UNIX CRAY2 supercomputer. Data files created by RETRO can be used as input for graphical software packages. Also, files containing experimental data, if properly formatted, can be used as input to other programs.

The purpose in writing DISPLAY was to present the theoretical data produced by RETRO in a variety of ways. DISPLAY is to aid in analyzing theoretical and experimental data, and to help in choosing appropriate wavelengths for ellipsometric study of various background and background-analyte surface scenarios. DISPLAY allows for direct comparison of experiment to theory.

DISPLAY uses the plotting package Disspla, version 10.0. Local printing capability is required for interactive work. All calls to Disspla are confined to subroutines that perform specific tasks. These subroutines print headings, draw axes, draw curves, etc. To change plotting packages, these routines (in the Disspla source code) need to be rewritten. Appendix V provides DISPLAY's source code, startup procedure, menu options, plot directives and data analysis options. Its source code is also provided in Appendix V.

We end this section by including Full Wave data from dry and wetted clay surfaces of variable roughness via execution of RETRO and DISPLAY programs. The clay sample is an admixture of three minerals in one-third proportion by weight; colloidal montmorillonite, kaolin, and illite. The optical constants and other information on how this clay pellet was fabricated are given in Reference 21. Figure 14a is the clay's calculated  $f_{11}(\lambda, \theta)$  element within a backscattering angle range of  $0 \leq \theta \leq 88^\circ$ , and wavelength band of  $9.0 \leq \lambda \leq 12.2 \mu m$ . Mean squared height of the clay pellet's surface is  $5.0 \mu m^2$  (smooth), its mean squared slope is 0.05, and the probability density functions of heights and slopes used in the theoretical model are Gaussian.<sup>30,31,32,33</sup> The three maxima arising in the matrix element-surfaces result from Reststrahlen absorption (a narrow wavelength region where a sharp jump occurs in the imaginary part of the complex refractive index in some minerals) in the soil material. In Figures 14b-f, respectively, mean square slope  $\langle \sigma^2 \rangle$  of the soil surface is increased in order from 0.05 to 0.10, 0.50, 1.00, 1.50, and finally to 2.00 while  $\langle h^2 \rangle$  is held constant. Recall that the  $f_{11}$  element is a measure of scattering power, as such, the pattern depicted in Figures 14a-f (i.e., scattering from a smooth to a rough surface as  $\langle \sigma^2 \rangle$  increases) is intuitively correct. At small slopes, the surface is spatially slow-varying and therefore most scattered energy occurs at  $0^\circ$  specular angle. As the soil surface slope and mean height go to zero, the Matrix element-surface of Figure 14a should reduce to the Fresnel reflection curve at 0 degrees, zero everywhere else. We see from Figures 14a-f, that as surface slopes increase (sharper topographical detail), scattering becomes more Lambertian-like, i.e., scattering energy becomes isotropic as shown by the increasing and broadening of the Mueller element surface  $f_{11}(\lambda, \theta)$  for angles beyond normal incidence ( $0^\circ$  degree), at the expense of decreasing specularly-reflected energy. We also note an intriguing result: the Reststrahlen peaks shift toward higher angle as roughness increases.

This same graphical analysis was conducted on the  $f_{21}$  element, results of which are presented in Figures 15a-f. Notice a trend in this Mueller element as the soil-surface roughness increases. The three Reststrahlen bands first appear positive, damp, reverse sign and decrease negative with increasing  $\langle \sigma^2 \rangle$ . The rate of change in the  $f_{21}$  bandhead amplitudes seems more rapid at the largest slopes. This change of sign in the absorption bands in the [2,1] element results from a changing relationship between horizontal and vertical polarized components of backscattered light when going from smooth to rough surfaces, an anomaly of Full-Wave theory<sup>30,31,32</sup>.

Our computer animation and visualization of the elements from a comprehensive Full Wave data bank show that  $\langle h^2 \rangle$  is an insensitive parameter to change in all IR Mueller element signatures between 10 - 100  $\mu m^2$ .

Mueller Element  $F_{11}$   
 composite Gaussian  
 $\langle h^2 \rangle = 5.000 \mu m^2$      $\sigma_s^2 = 0.050$

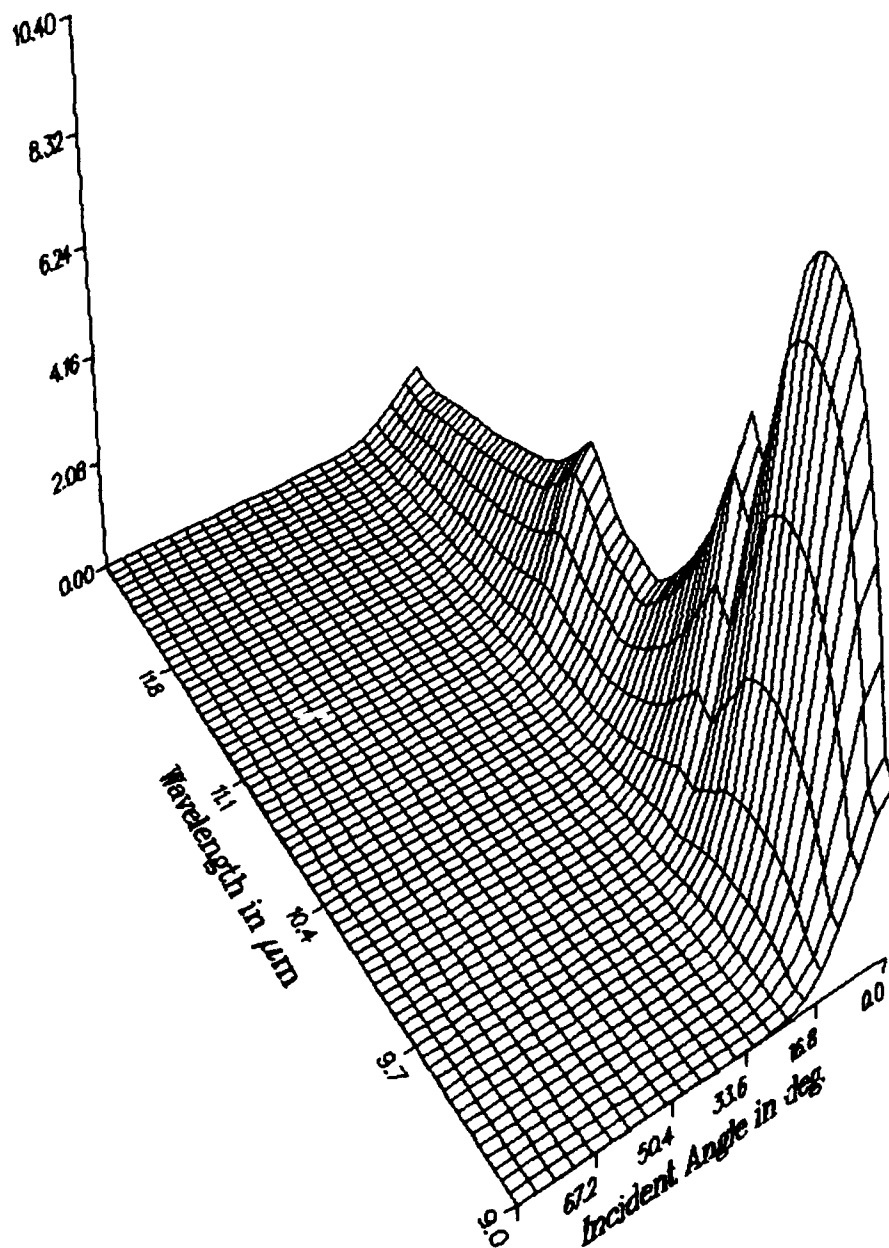


Figure 14a. Full-Wave model prediction of the Mueller matrix element  $f_{11}$  in the backscattering direction from a composite soil material assuming a surface structure of Gaussian distributed slopes ( $\sigma$ ) and heights ( $h$ ). The topographical mean-squared surface height (in  $\mu m^2$ ) and slope of the soil sample in (a) are, respectively, 5.0 and 0.05. Note that as  $\langle \sigma^2 \rangle$  increases in the following five figures b-f, the surface becomes a more Lambertian-like (isotropic) reflector as expected. The model restricts  $\langle \sigma^2 \rangle \leq 2$ .

Mueller Element  $F_{11}$   
 composite Gaussian  
 $\langle h^2 \rangle = 5.000 \mu m^2$      $\sigma_s^2 = 0.100$

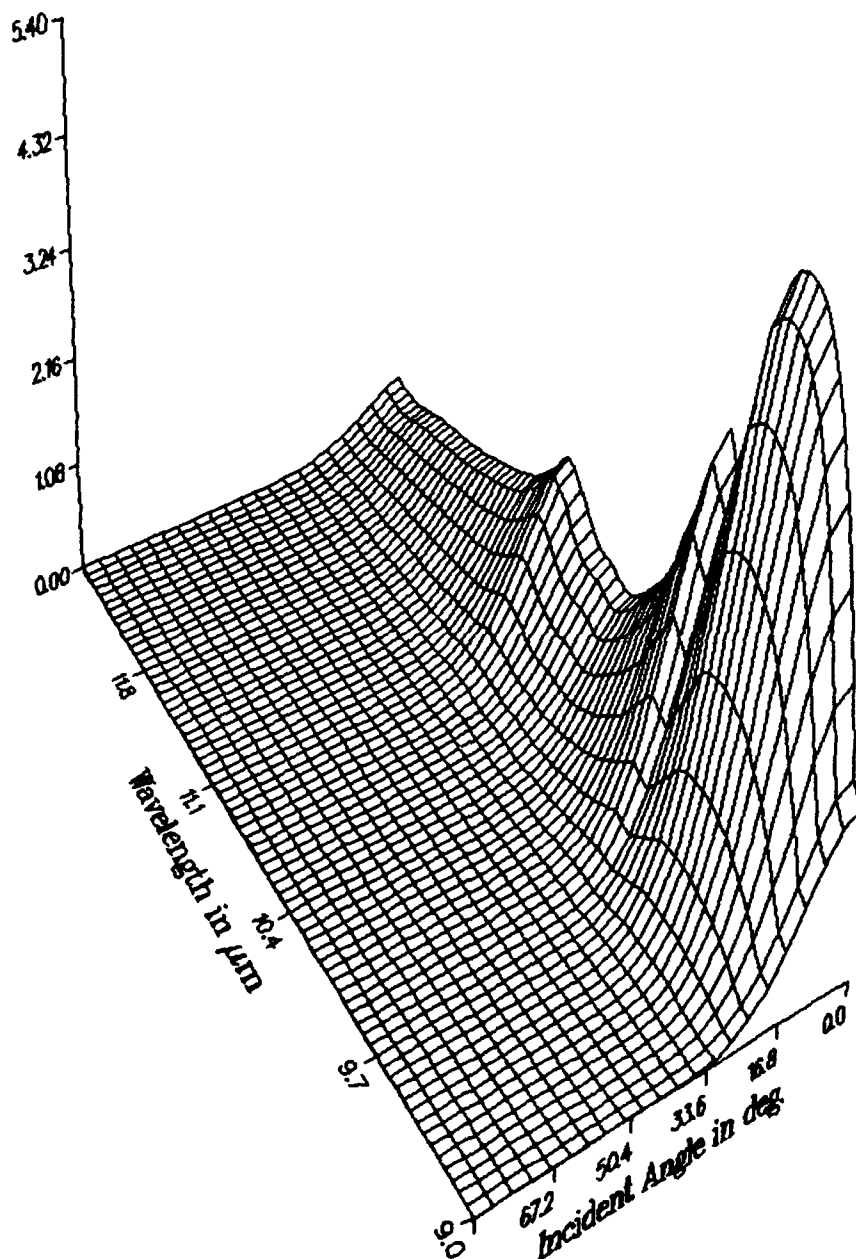


Figure 14b. Full-Wave model prediction of the Mueller matrix element  $f_{11}$  in the backscattering direction from a composite soil material assuming a surface structure of Gaussian distributed slopes ( $\sigma$ ) and heights ( $h$ ). The topographical mean-squared surface height (in  $\mu m^2$ ) and slope of the soil sample in (a) are, respectively, 5.0 and 0.10.

Mueller Element  $F_{11}$   
 composite Gaussian  
 $\langle h^2 \rangle = 5.000 \mu m^2$      $\sigma_s^2 = 0.500$

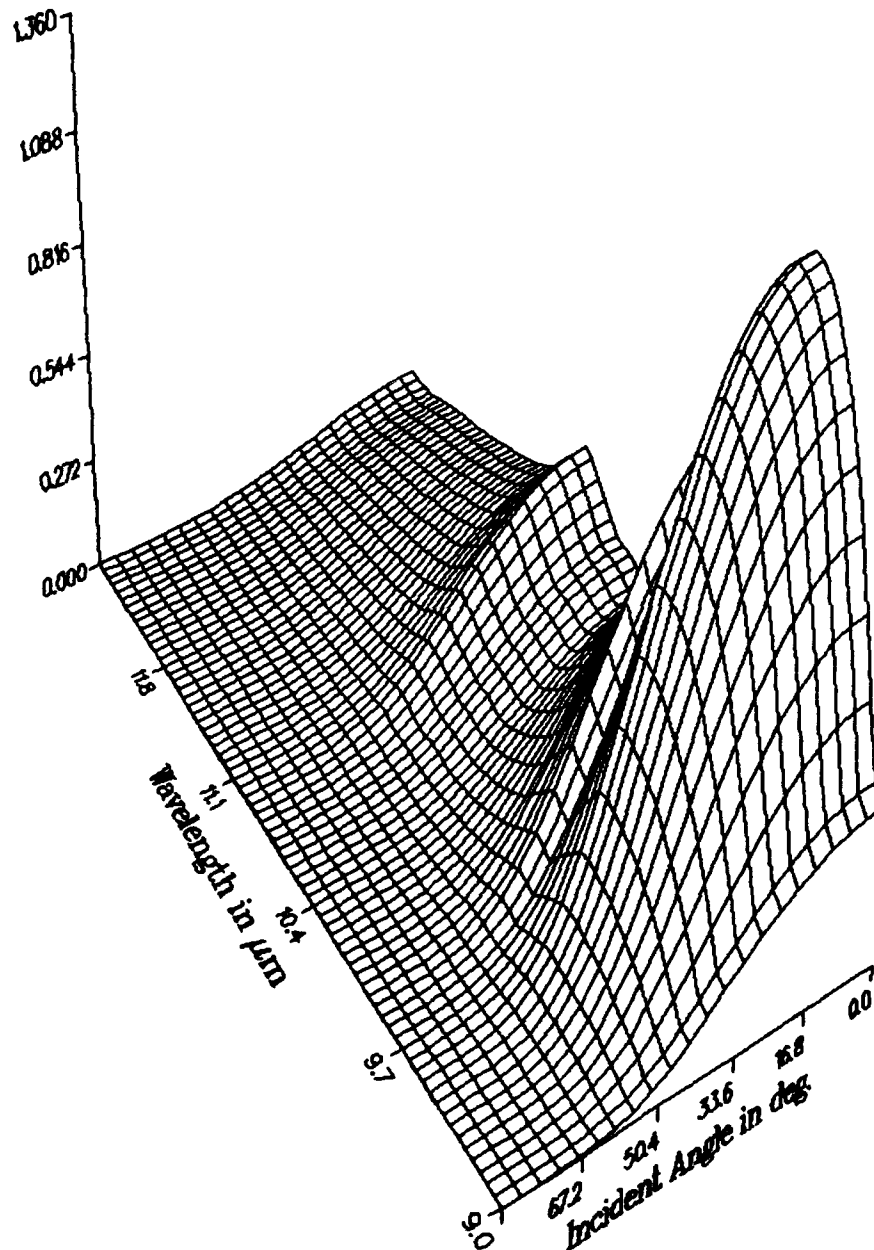


Figure 14c. Full-Wave model prediction of the Mueller matrix element  $f_{11}$  in the backscattering direction from a composite soil material assuming a surface structure of Gaussian distributed slopes ( $\sigma$ ) and heights ( $h$ ). The topographical mean-squared surface height (in  $\mu m^2$ ) and slope of the soil sample in (a) are, respectively, 5.0 and 0.50.

Mueller Element  $F_{11}$   
 composite Gaussian  
 $\langle h^2 \rangle = 5.000 \mu m^2$      $\sigma_s^2 = 1.000$

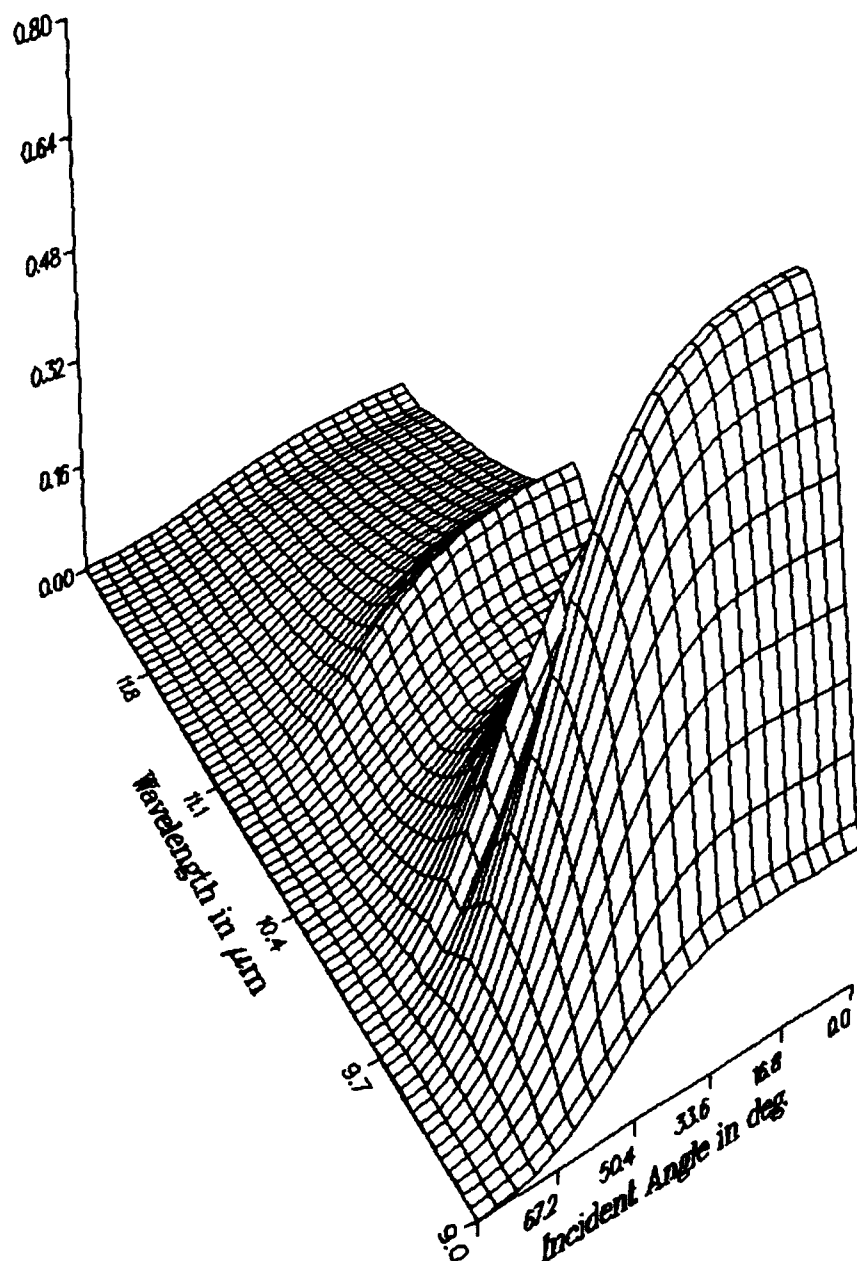


Figure 14d. Full-Wave model prediction of the Mueller matrix element  $f_{11}$  in the backscattering direction from a composite soil material assuming a surface structure of Gaussian distributed slopes ( $\sigma$ ) and heights ( $h$ ). The topographical mean-squared surface height (in  $\mu m^2$ ) and slope of the soil sample in (a) are, respectively, 5.0 and 1.00.

Mueller Element  $F_{11}$   
 composite Gaussian  
 $\langle h^2 \rangle = 5.000 \mu m^2$      $\sigma_s^2 = 1500$

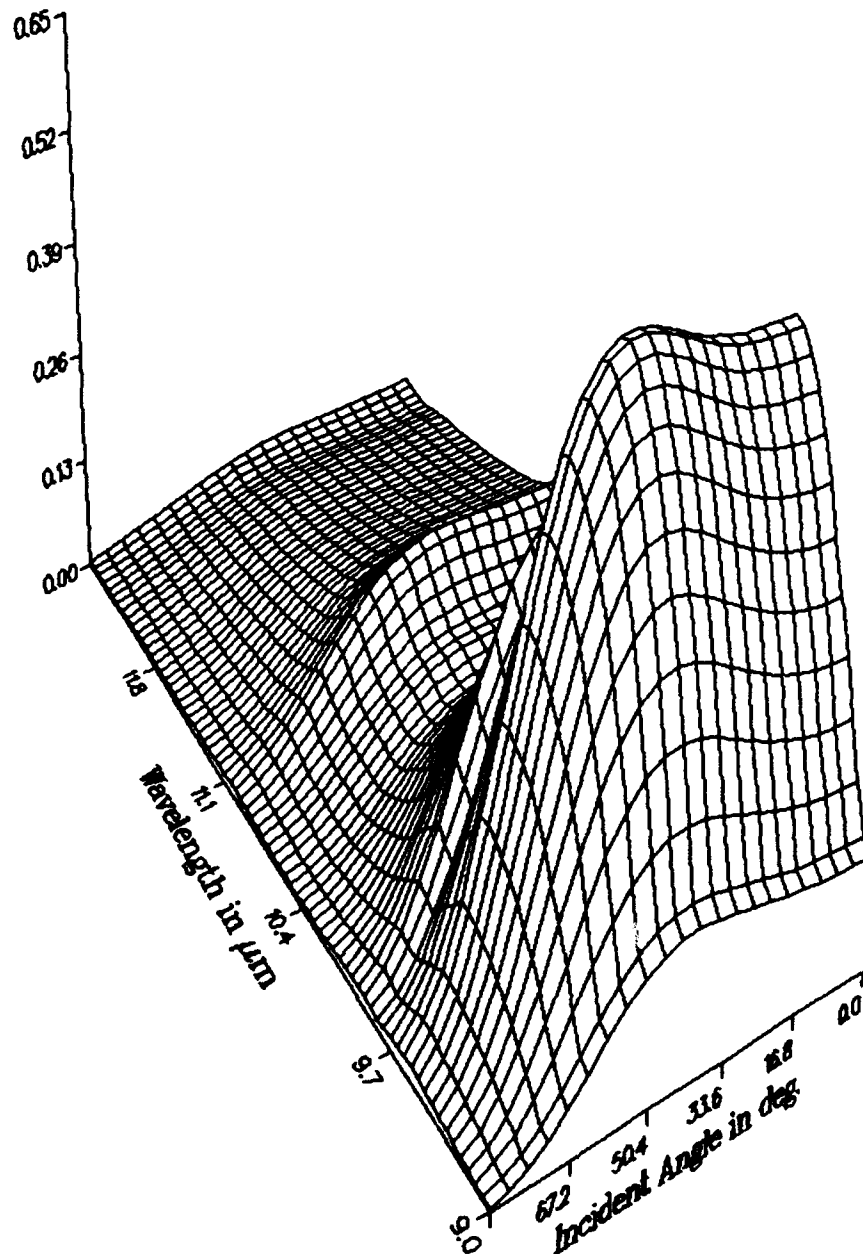


Figure 14e. Full-Wave model prediction of the Mueller matrix element  $f_{11}$  in the backscattering direction from a composite soil material assuming a surface structure of Gaussian distributed slopes ( $\sigma$ ) and heights ( $h$ ). The topographical mean-squared surface height (in  $\mu m^2$ ) and slope of the soil sample in (a) are, respectively, 5.0 and 1.50.

Mueller Element  $F_{11}$   
 composite Gaussian  
 $\langle h^2 \rangle = 5.000 \mu m^2$   $\sigma_s^2 = 2.000$

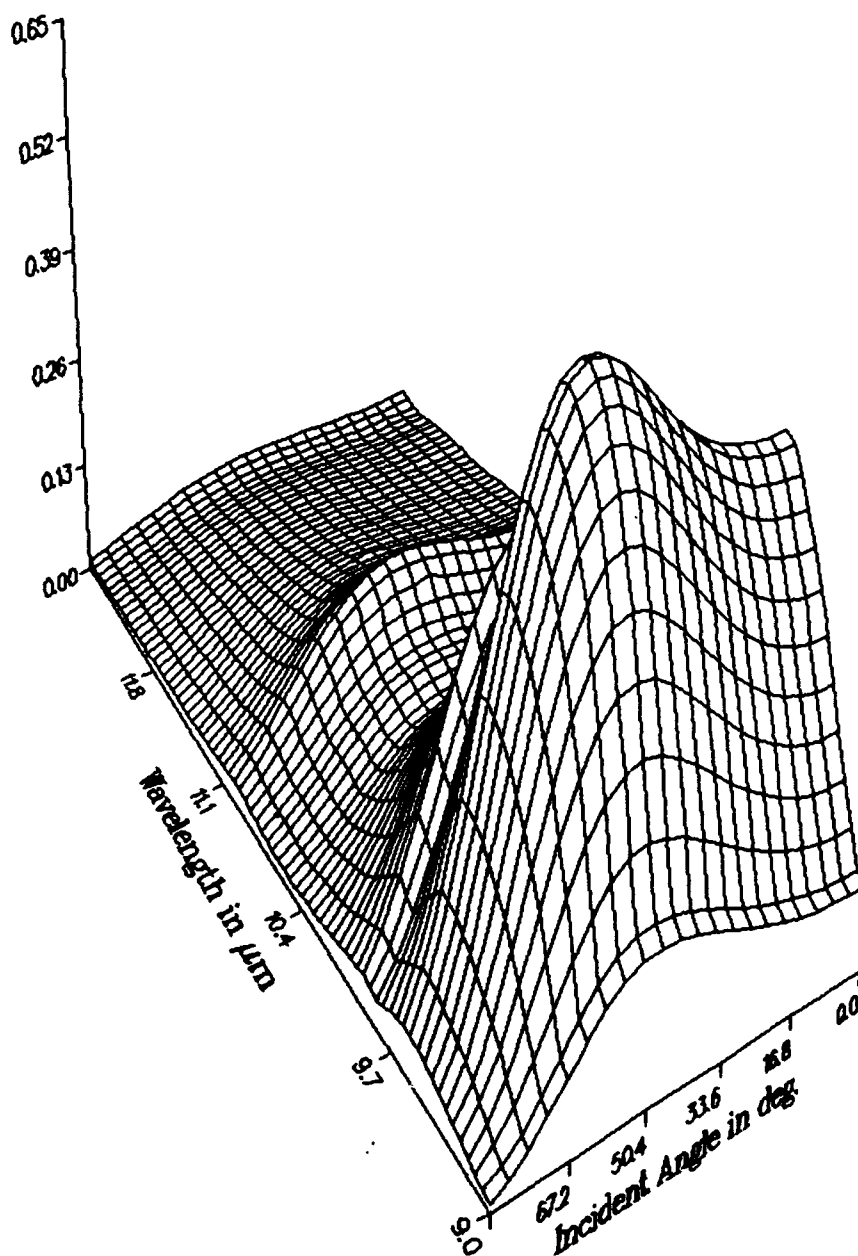


Figure 14f. Full-Wave model prediction of the Mueller matrix element  $f_{11}$  in the backscattering direction from a composite soil material assuming a surface structure of Gaussian distributed slopes ( $\sigma$ ) and heights ( $h$ ). The topographical mean-squared surface height (in  $\mu m^2$ ) and slope of the soil sample in (a) are, respectively, 5.0 and 2.00.

Mueller Element  $F_{12}$   
 composite Gaussian  
 $\langle h^2 \rangle = 5.000 \mu m^2$      $\sigma_s^2 = 0.050$

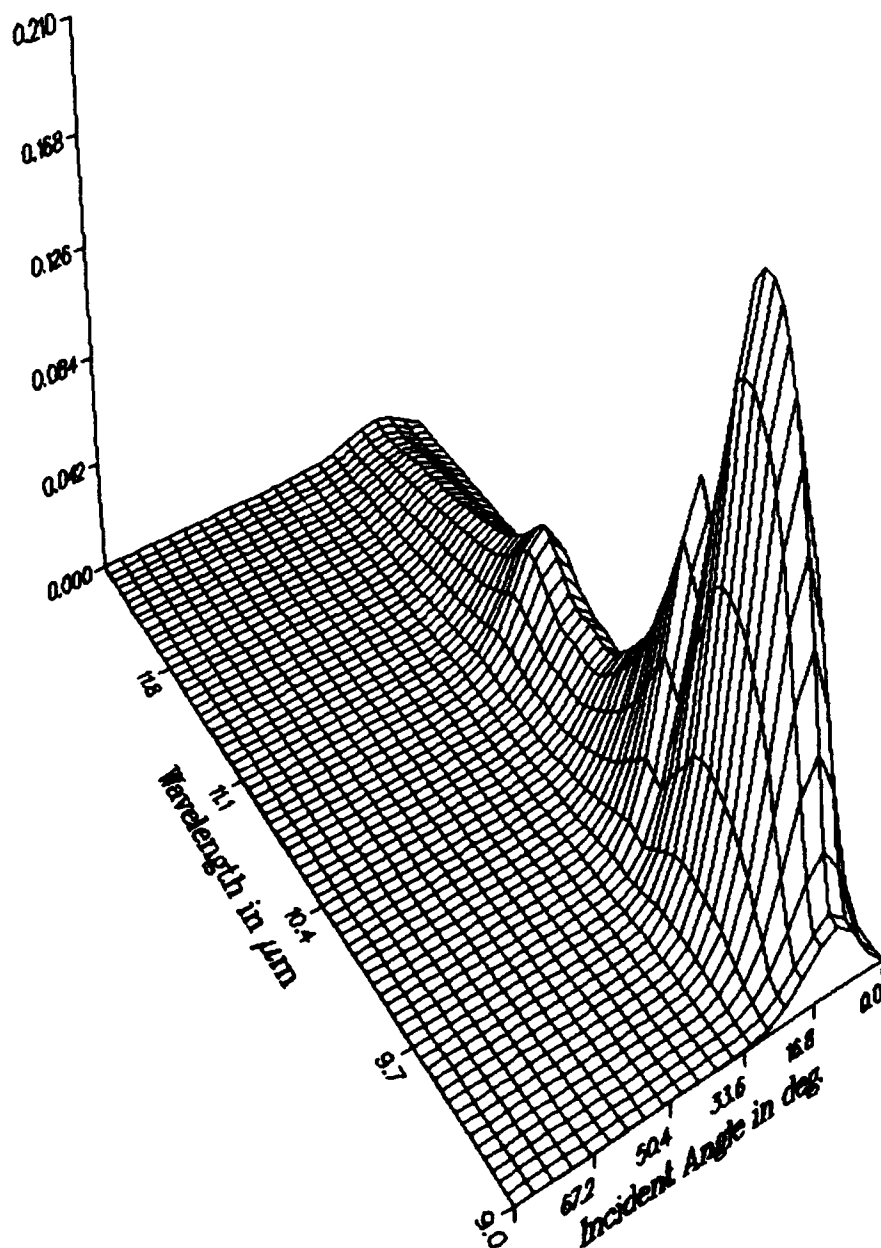


Figure 15a. Full-Wave model prediction of the Mueller matrix element  $f_{12}$  in the backscattering direction from a composite soil material assuming a surface structure of Gaussian distributed slopes ( $\sigma$ ) and heights ( $h$ ). The topographical mean-squared surface height (in  $\mu m^2$ ) and slope of the soil sample in (a) are, respectively, 5.0 and 0.05. Note the sign reversal in the material's bandheads as  $\langle \sigma^2 \rangle$  increases in the following figures b-e. The model restricts  $\langle \sigma^2 \rangle \leq 2$ .

Mueller Element  $F_{12}$   
 composite Gaussian  
 $\langle h^2 \rangle = 5.000 \mu m^2$   $\sigma_s^2 = 0.100$

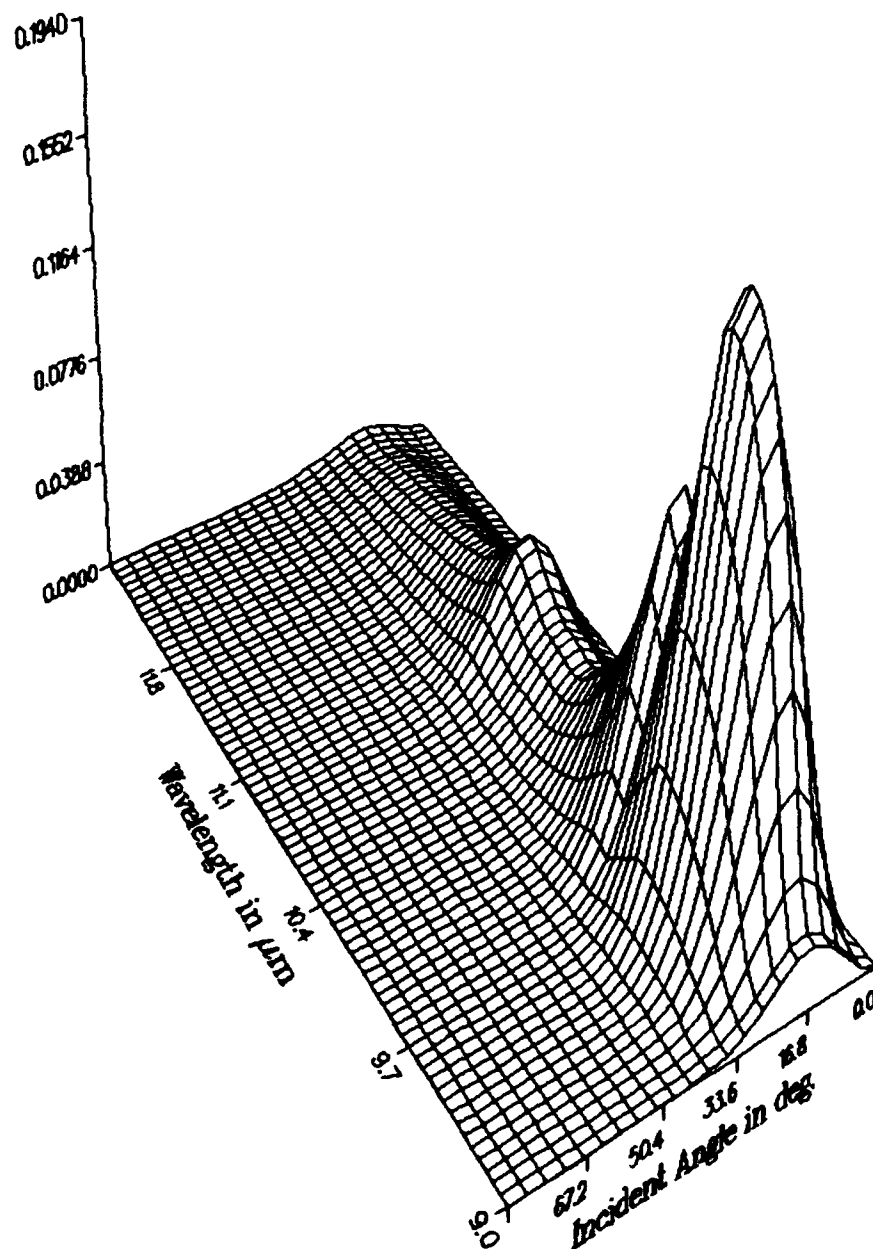


Figure 15b. Full-Wave model prediction of the Mueller matrix element  $f_{12}$  in the backscattering direction from a composite soil material assuming a surface structure of Gaussian distributed slopes ( $\sigma$ ) and heights ( $h$ ). The topographical mean-squared surface height (in  $\mu m^2$ ) and slope of the soil sample in (a) are, respectively, 5.0 and 0.10.

Mueller Element  $F_{12}$   
 composite Gaussian  
 $\langle h^2 \rangle = 5.000 \mu m^2$      $\sigma_s^2 = 0.500$

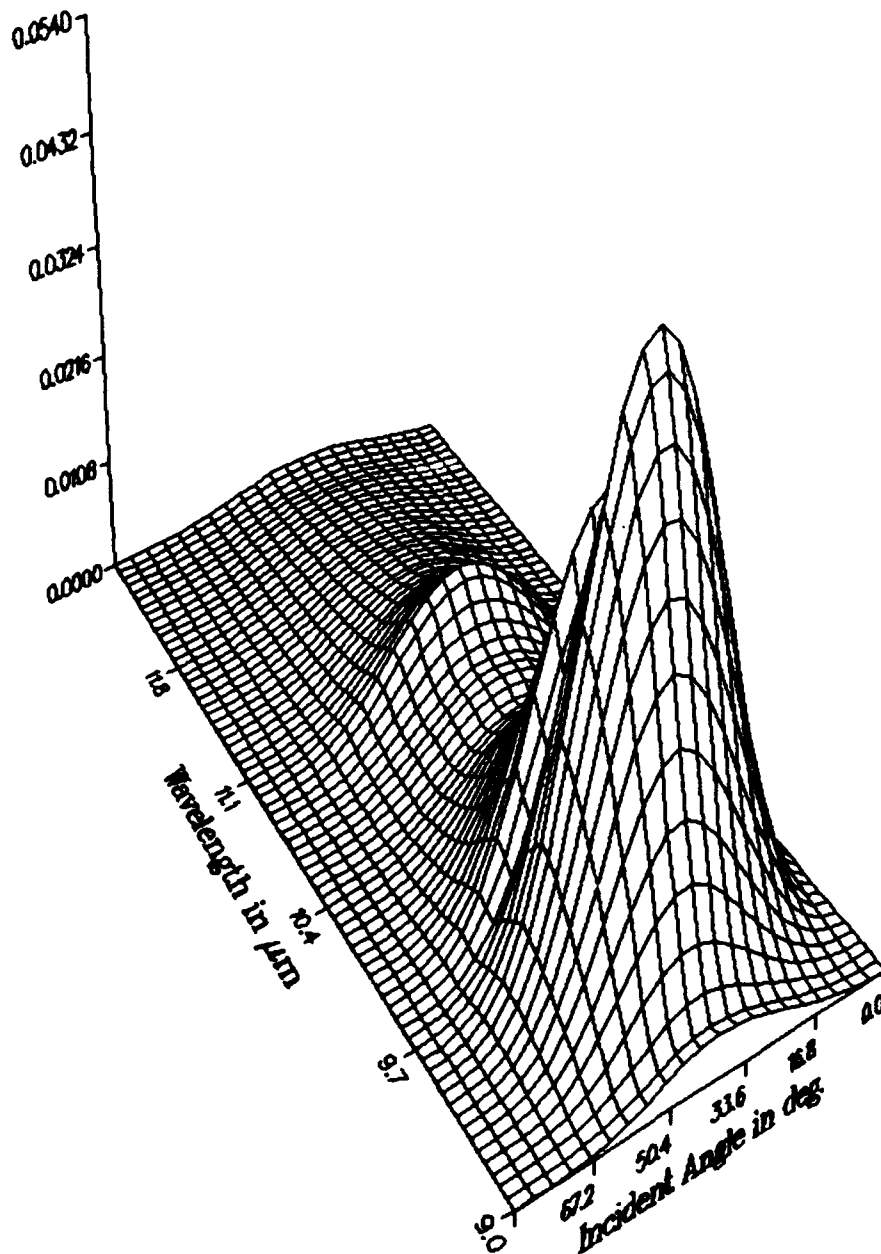


Figure 15c. Full-Wave model prediction of the Mueller matrix element  $f_{12}$  in the backscattering direction from a composite soil material assuming a surface structure of Gaussian distributed slopes ( $\sigma$ ) and heights ( $h$ ). The topographical mean-squared surface height (in  $\mu m^2$ ) and slope of the soil sample in (a) are, respectively, 5.0 and 0.50.

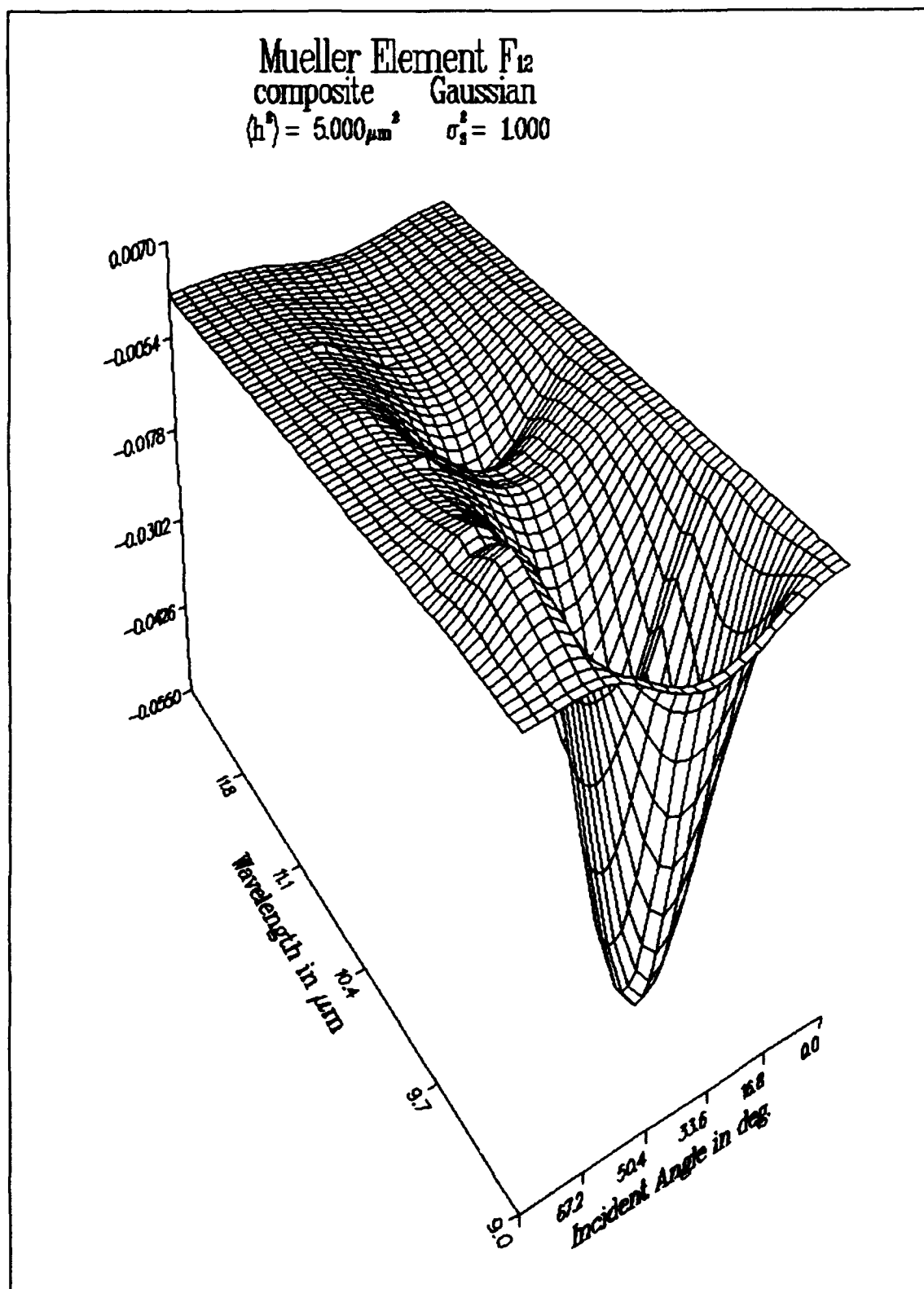


Figure 15d. Full-Wave model prediction of the Mueller matrix element  $f_{12}$  in the backscattering direction from a composite soil material assuming a surface structure of Gaussian distributed slopes ( $\sigma$ ) and heights ( $h$ ). The topographical mean-squared surface height (in  $\mu m^2$ ) and slope of the soil sample in (a) are, respectively, 5.0 and 1.00.

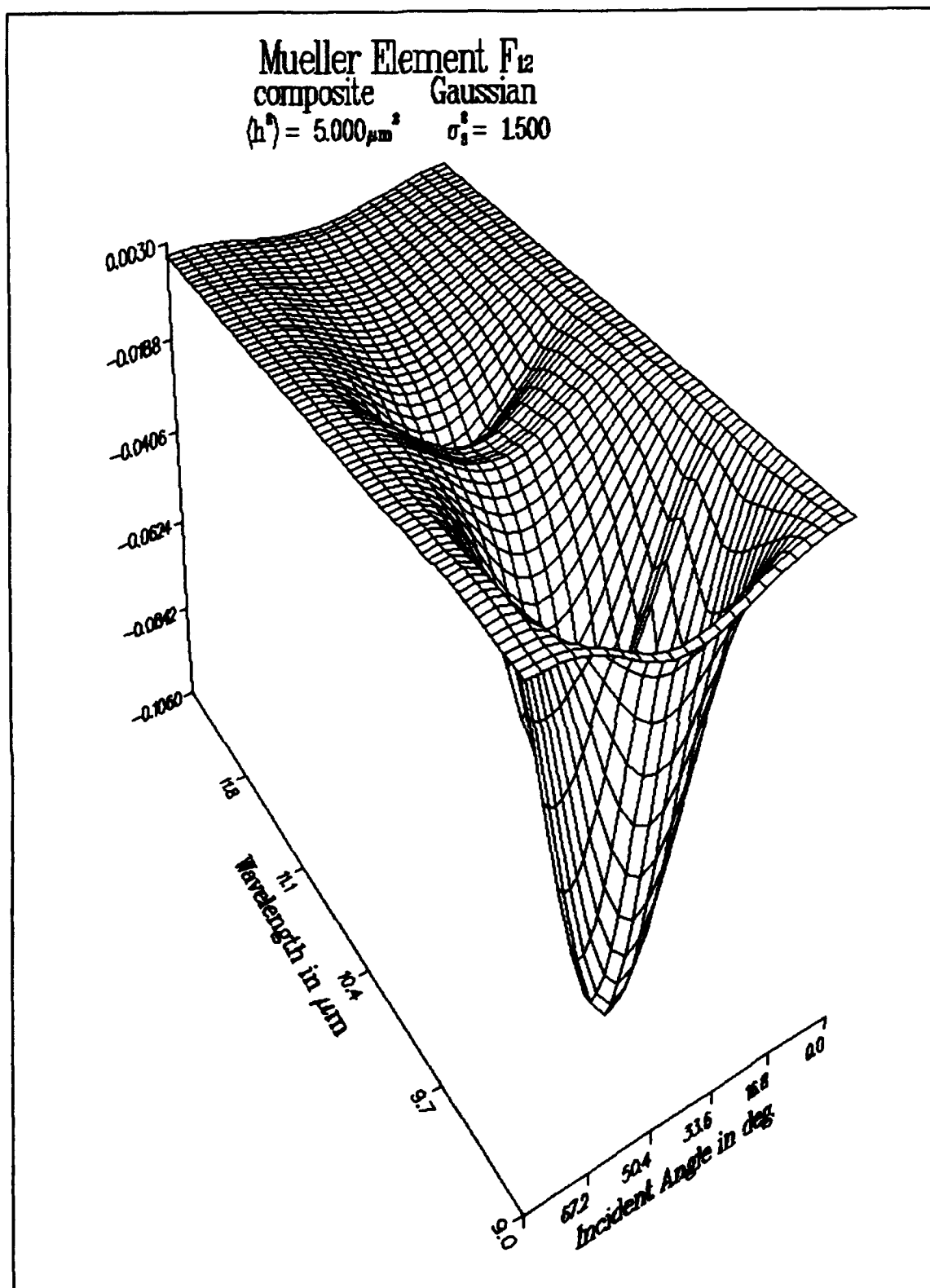


Figure 15e. Full-Wave model prediction of the Mueller matrix element  $f_{12}$  in the backscattering direction from a composite soil material assuming a surface structure of Gaussian distributed slopes ( $\sigma$ ) and heights ( $h$ ). The topographical mean-squared surface height (in  $\mu m^2$ ) and slope of the soil sample in (a) are, respectively, 5.0 and 1.50.

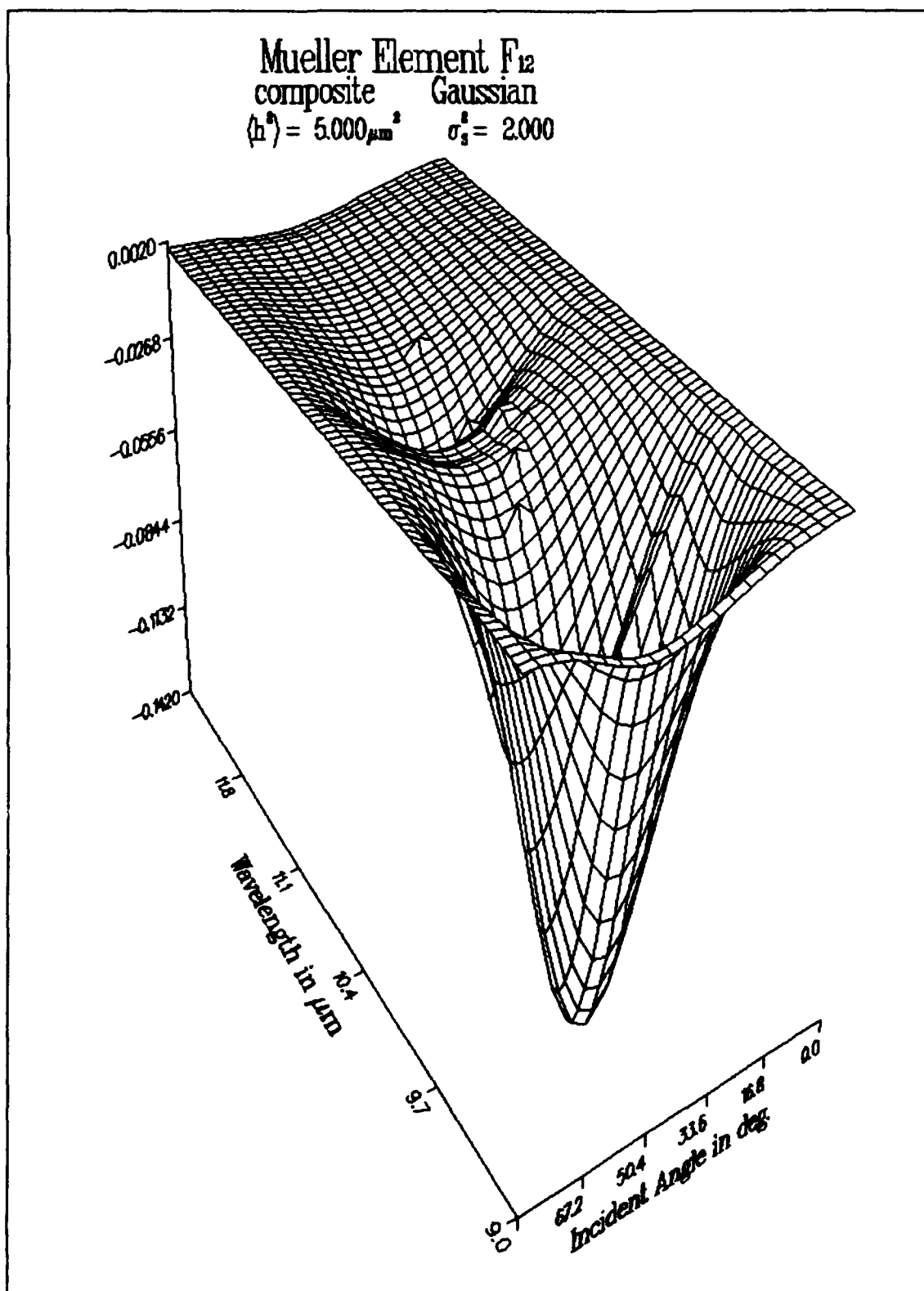


Figure 15f. Full-Wave model prediction of the Mueller matrix element  $f_{12}$  in the backscattering direction from a composite soil material assuming a surface structure of Gaussian distributed slopes ( $\sigma$ ) and heights ( $h$ ). The topographical mean-squared surface height (in  $\mu m^2$ ) and slope of the soil sample in (a) are, respectively, 5.0 and 2.00.

An example of how one could choose laser beam angles and wavelengths, per Mueller element, for detecting contaminants DMMP, DIMP and SF96 on a soil surface is illustrated in Figures 16 through 18. Contours  $F(\lambda, \theta)$  in the  $\lambda - \theta$  plane are plotted for Mueller elements  $f_{11}$ ,  $f_{12}$ , and  $f_{34}$  for a clay surface with  $\langle h^2 \rangle = 20 \mu m^2$ , and  $\sigma_s^2 = 0.5$ , and elements of the same surface coated by SF96, DMMP, and DIMP contaminant liquids. (The assumption here is that after the soil is wetted by the liquid contaminant, its surface becomes uniformly coated and conforms to the unwetted soil surface geometry. Also, the coating is optically thick.) The cross-hatched sections in these figures are  $(\theta, \lambda)$  regions where the analyte can be discerned from the soil background. These regions are set subtractions of data from dry and contaminated soil, and are clearly contrasted in the Mueller elements. The programs RETRO/DISPLAY (Sections 5.2.1-5.2.2, and Appendix V) were executed in producing these data. In the  $f_{11}$  element, SF96- and DMMP-contaminated soils yield predominate analyte signals at the higher wavelengths starting  $\approx 12.2 \mu m$ , and at angles not exceeding  $48^\circ$ , while DIMP cannot be distinguished in this Mueller element. In element  $f_{12}$ , Figure 17, SF96- and DMMP-contaminated surfaces are still disjoint from the dry soil surface at the higher wavelengths, but their detection  $\lambda - \theta$  domain is more restrictive in angle. DIMP still cannot be detected via  $f_{12}$ . Finally, in Figure 18, we see that DIMP can be detected via  $f_{34}$  in the cross-hatched region near  $10.15 \mu m$  and  $12^\circ - 48^\circ$ , as are SF96 and DMMP in the higher  $\lambda - \theta$  domains.

This kind of graphical analysis can be extended to the remaining independent Mueller elements, and a selection of angles and wavelengths can likely be chosen to discriminate against and between a vast group of chemical contaminants that exhibit an IR vibrational spectrum. Moreover, through experiment and theoretical modelling, we hope to establish how these domain patterns change with target concentration, allowing one to map the contaminant once detection is established, and monitor its fate.

# Mueller Element $F_{11}$ Comparison of Materials

$1.950 \times 10^{-1}$   
 $5.850 \times 10^{-1}$   
 $9.750 \times 10^{-1}$   
 $1.365 \times 10^0$   
 $1.755 \times 10^0$

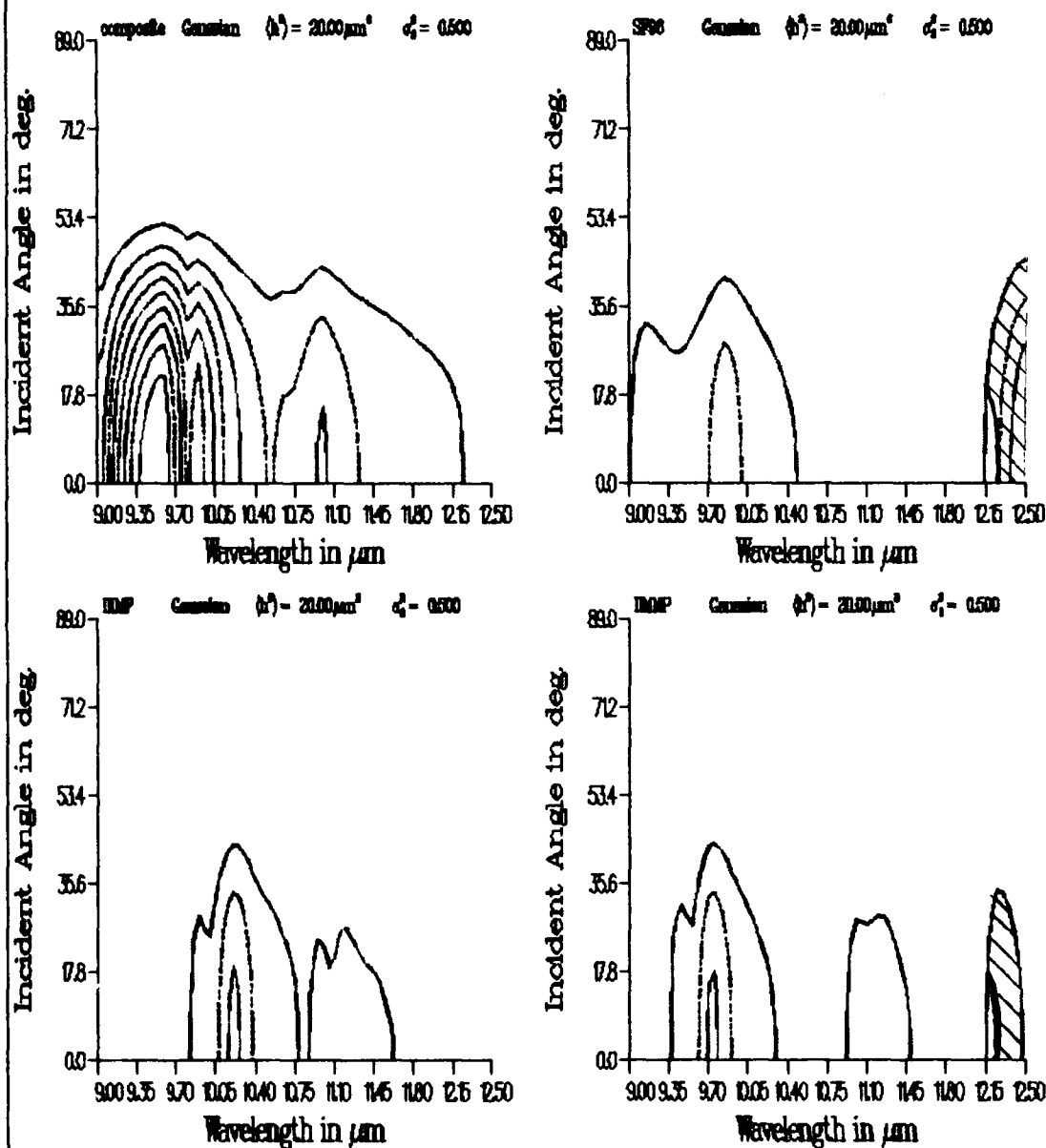


Figure 16. Regions of irradiation wavelength and backscattering angle in the  $f_{11}$  Mueller matrix element that are most useful for discriminating against liquid chemical agent simulants SF96, DIMP, and DMMP on a soil surface. The crosshatched wavelength-angle domains are areas where the ellipsometer sensor should be set to, so that a signal from the contaminants can be detected. These regions of maximum analyte detections result from a subtraction of dry- from wet-soil data sets. Note that DIMP cannot be detected in  $f_{11}$ .

# Mueller Element $F_{12}$ Comparison of Materials

$1.800 \times 10^{-2}$   
 $7.400 \times 10^{-2}$   
 $1.300 \times 10^{-1}$   
 $1.860 \times 10^{-1}$   
 $2.420 \times 10^{-1}$

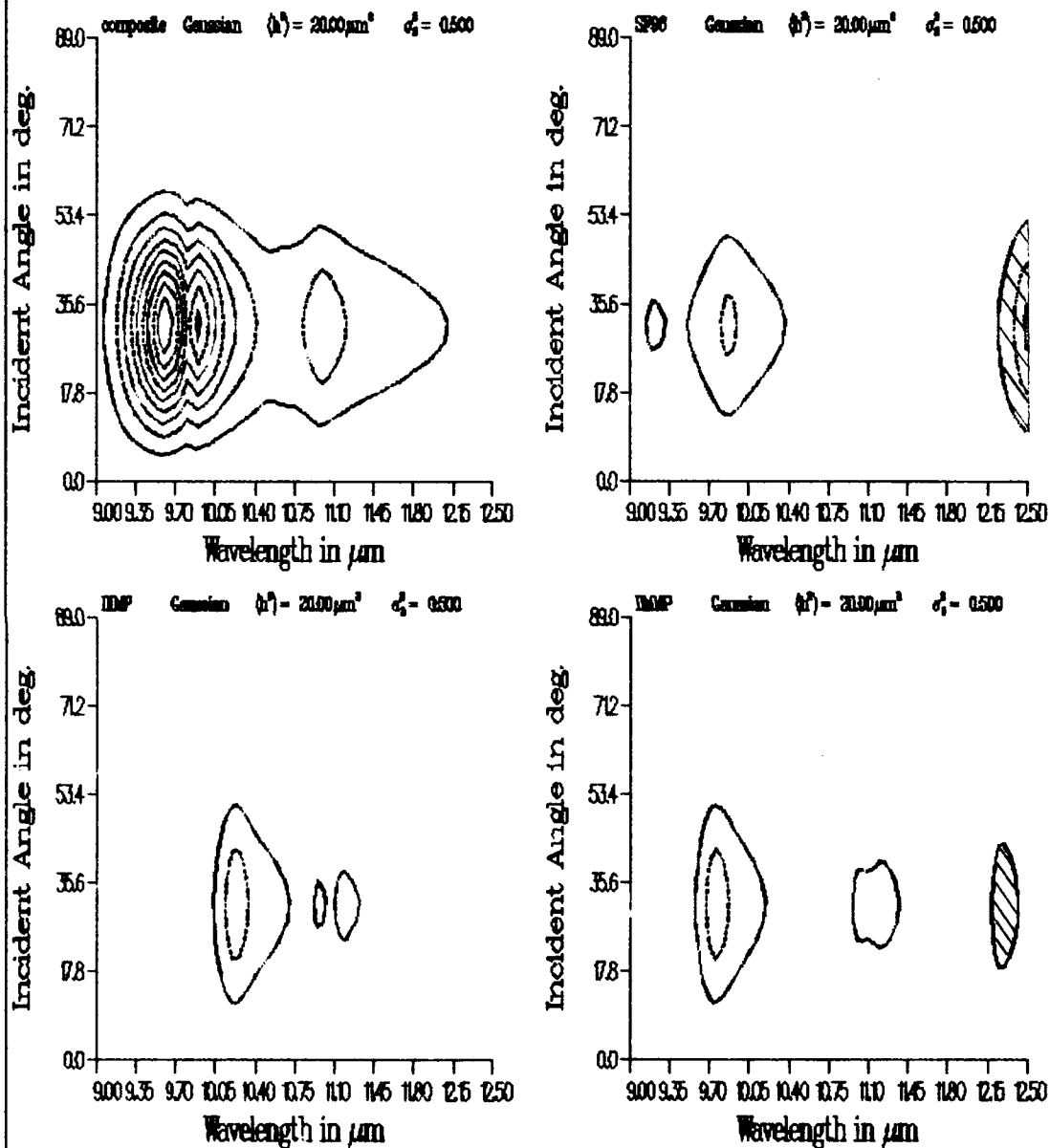


Figure 17. Regions of irradiation wavelength and backscattering angle in the  $f_{12}$  Mueller matrix element that are most useful for discriminating against liquid chemical agent simulants SF96, DIMP, and DMMP on a soil surface. The crosshatched wavelength-angle domains are areas where the ellipsometer sensor should be set to, so that a signal from the contaminants can be detected. These regions of maximum analyte detections result from a subtraction of dry- from wet-soil data sets. Note that DIMP cannot be detected in  $f_{12}$ .

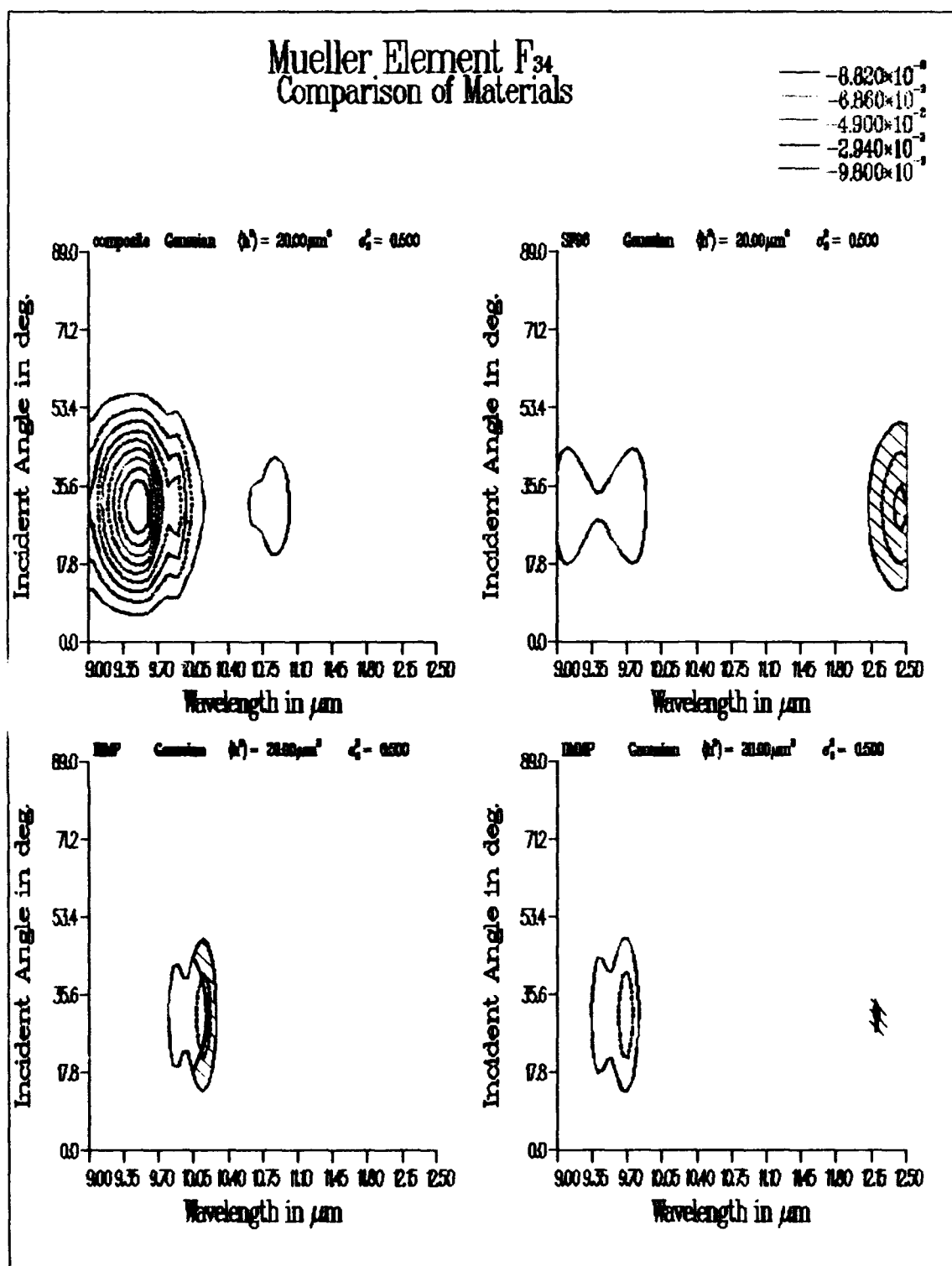


Figure 18. Regions of irradiation wavelength and backscattering angle in the  $f_{34}$  Mueller matrix element that are most useful for discriminating against liquid chemical agent simulants SF96, DIMP, and DMMP on a soil surface. The crosshatched wavelength-angle domains are areas where the ellipsometer sensor should be set to, so that a signal from the contaminants can be detected. These regions of maximum analyte detections result from a subtraction of dry- from wet-soil data sets. Note that all three simulants can be detected in  $f_{34}$ .

### 5.2.3 DETECT: Remote Detection Application of Full Wave Theory.

Program DETECT, written by Mark Haugland, is a detection algorithm for this multi-wavelength, two-modulator IR ellipsometer. The program serves 2 purposes: (1) to locate optimum angles of incidence and wavelengths for use in discriminating between a contaminated and a dry surface; and (2) to identify those Mueller matrix elements that can be used to discriminate between the analyte (contaminant) and the background (substrate) at optimum angles of incidence and laser wavelengths.

The Mueller matrix for each scattering surface is a function of wavelength, incident angle, mean square height, and mean square slope. One way to select useful combinations of incident angle and wavelength is to, as described in the previous section, visually inspect DISPLAY plots of the difference of the Mueller matrices for both contaminated and bare materials. Due to the sheer amount of data involved, this method is time consuming and most difficult when extracting quantitative information quickly.

The most current version of Full Wave theory can compute Mueller matrices for stratified media with an optically thick contaminant layer, i.e., one rough interface is considered. Given a layered boundary value problem, the backscattering Mueller matrix  $F$  for one or more randomly rough interfaces has 6 linearly independent entries, given the assumptions and restrictions on the media prescribed previously in the Full Wave model. These matrix elements are used in the construction of a 6-dimensional vector  $v$  in the following manner.

$$\bar{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} \quad (32)$$

$$v_1 = f_{11}, \quad v_2 = f_{12}, \quad v_3 = f_{22}, \quad v_4 = f_{33}, \quad v_5 = f_{34}, \quad v_6 = f_{44}. \quad (33)$$

Let  $\bar{v}^b$  represent the background material, and  $\bar{v}^t$  represent the target material. As a first step, consider a vector  $\bar{d}$  defined by

$$\bar{d} = \bar{v}^t - \bar{v}^b. \quad (34)$$

The magnitude of  $\bar{d}$  is given by

$$|\bar{d}| = \left( \sum_{k=1}^6 d_k^2 \right)^{\frac{1}{2}}. \quad (35)$$

The magnitude of Equation 35 is the first step in selecting a useful detection discriminant (combination of incident angle and wavelength). Terms in Equation (35) with  $|v_k^b| > |v_k^i|$  are excluded, i.e., relatively strong returns from the background material filtered (similar to the subtracted data sets of Section 5.2.2). The following equation uses step functions to disregard relatively strong returns from the background material.

$$x_k = d_k u(|v_k^i| - |v_k^b|) \quad (36)$$

Here,  $u(\cdot)$  is a unit step function. Candidate incident angle and wavelength detection parameters are found by calculating

$$|\bar{x}| = \left( \sum_{k=1}^6 x_k^2 \right)^{\frac{1}{2}} \quad (37)$$

and enforcing the condition  $|\bar{x}| > |\bar{x}|_{\max}$ . The useful Mueller matrix detection elements at these wavelength and incident angle pairs correspond to the nonzero components of  $\bar{x}$ .

Consider cases where  $x_k \gg x_j$ . Often, this results when  $v_k^i$  and  $v_k^b$  are large in magnitude compared to  $v_j^i$  and  $v_j^b$ . Consequently,  $x_j$  has negligible effect on the value of Equation (37) even though  $v_j^i$  and  $v_j^b$  may exhibit strikingly different behavior. Scaling each component of  $x$  by an appropriate factor will yield a test producing results that depend on the relative size of two matrix elements rather than the magnitude of their difference. The vector  $y$  has this property

$$y_k = \frac{x_k}{\sqrt{|v_k^i v_k^b|}}. \quad (38)$$

As in Equation (37), combinations of incident angle and wavelength of particular interest may be found by computing

$$|\bar{y}| = \left( \sum_{k=1}^6 y_k^2 \right)^{\frac{1}{2}} \quad (39)$$

and requiring  $|\bar{y}| > |\bar{y}|_{\max}$ . Again, useful Mueller matrix elements correspond to the non-zero components of  $y$ .

Should a matrix element for either the target or the background tend toward zero, Equation (39) becomes singular (tends to infinity). For this reason, it is necessary to set  $y_k$  equal to zero in the DETECT program whenever  $v_k^b$  or  $v_k^t$  are zero.

The refractive index  $n(\lambda) - ik(\lambda)$  of a material is a complex function of wavelength which plays an instrumental role in determining a material's response to an incident photon. Peaks in the Mueller matrix elements usually occur at resonant wavelengths. Resonant wavelengths correspond to local maxima in  $k$ .

DETECT identifies all resonant wavelengths for both the background and the target materials. The program also identifies all local minima in  $k$ . DETECT helps the user identify the correlation between on- and off-resonance wavelengths and numerical results from Equations (36), (37), (38), and (39).

We have thus far developed the criterion for finding useful combinations of incident angle, wavelength, per Mueller matrix element susceptible to the analyte. We have not yet accounted for the variational error of parameters expected in experimental operation. Inputting several sets of experimental data to the current version of DETECT, computing average values for  $\bar{v}^b$  and  $\bar{v}^t$ , and dividing each component of these vectors by their respective variances may account for variability in the experimental measurements.

The remainder of this section discusses a more sophisticated way of accounting for experimental variability than averaging data and dividing by variances. We discuss simulation of experimental uncertainties with the theoretical data base. First, however, we review the definitions of variance, covariance, and the covariance matrix.

The variance of a univariate quantity  $z$  is defined by

$$\sigma_z = \frac{1}{N} \sum_{i=1}^N (z_i - z_{ave})^2 \quad (40)$$

where  $N$  is the number of samples taken and  $z_{ave}$  is the mean value of  $z$  over  $N$  samples.  $N$  should be large enough so that increasing it will not change  $\sigma_z$ . The expected or average value  $E(\bar{r})$  of a random vector  $\bar{r}$  is the vector whose components are the average value of each component of  $\bar{r}$ , that is:<sup>35</sup>

$$E(\bar{r}) = \begin{bmatrix} E(r_1) \\ E(r_2) \\ \vdots \\ E(r_k) \end{bmatrix} \quad (41)$$

When generalizing variance to multidimensional quantities, one defines the covariance of 2 components  $r_i$  and  $r_j$  of a random vector  $\bar{r}$  by:<sup>35</sup>

$$\text{cov}(r_i, r_j) = E [(r_i - E(r_i))(r_j - E(r_j))]. \quad (42)$$

In Equation (42), the covariance of  $r_i$  and  $r_j$  is the average of the product of  $r_i$ 's and  $r_j$ 's deviation from their respective mean values. For  $i=j$ , Equation (42) is the variance of  $r_j$ . If  $r_i$  and  $r_j$  are uncorrelated, then Equation (42) is identically zero. The covariance matrix contains the covariances of all components of  $r$ . The elements of the covariance matrix are arranged according to the following definition:

$$\Sigma = E [(\bar{r} - E(\bar{r}))(\bar{r} - E(\bar{r}))^T] \quad (43)$$

where symbol T denotes transposition. From Equation (43), it is clear that  $\Sigma_{ij} = \text{cov}(r_i, r_j)$  and that the covariance matrix is symmetric.

In the next section it is assumed that the covariance matrix is positive definite. This is necessary to insure that the quadratic forms in question are ellipsoidal.<sup>35</sup>

Hotelling's T-squared method is one way to check if a hypothesis is true or false. For this application, the first hypothesis is that a given backscatter angle/wavelength combination is useful. The second hypothesis is that the contaminant is present. Throughout this section, it is assumed that the covariance matrices for the contaminated and uncontaminated surfaces are equal.

This method uses the boundary of an ellipsoid as the test criterion. The ellipsoidal region is defined by

$$\bar{x}^T \Sigma^{-1} \bar{x} = c^2 \quad (44)$$

where  $c$  is a constant,  $\Sigma^{-1}$  is the inverse of the  $n$  dimensional covariance matrix for the data contained in  $x$ , and  $\bar{x}$  is a  $1 \times n$  column vector whose entries represent the average value of the quantity defined by Equation (35). Optimum angles and wavelength are those for which

$$c > c_{\max} \quad (45)$$

i.e.,  $\bar{x}$  lying outside of the ellipsoid defined by Equation (44). As earlier stated, useful Mueller matrix elements correspond to the non-zero components of  $\bar{x}$ .

Now that a set of useful angle-wavelength pairs have been found, it is time to use them to identify a contaminant. One way to accomplish this is to evaluate Equation (44) at several

\* In Equation (36),  $k$  is 6. Experimental results may show that there are more than 6 linearly independent Mueller matrix elements. For this reason,  $n$  is left as an unknown dimension  $\leq 16$ .

angles and wavelengths, say  $m$ , for an unknown sample, then store these values in the  $1 \times m$  column vector  $c$ . Let  $c_{ref}$  denote the value of  $c$  for an analyte on some surface. Define  $a$  by:

$$\bar{a} = \bar{c} - \bar{c}_{ref}. \quad (46)$$

The contaminant is present if  $|a| < |a|_{max}$ , and its concentration (mass density) is approximated by using  $c_{ref}$  representing various densities in Equation (46). The  $c_{ref}$  that results in the smallest  $|a|$  is the closest approximation to  $c$ . Hence, the unknown sample has approximately the same concentration as the known sample whose  $c_{ref}$  resulted in the smallest  $|a|$ .

Using Equation (46) to identify contaminants works in principle, but information is not used to a full extent in representing the Mueller matrix data collected at each angle and wavelength by a single scalar. This method may involve using more angle-wavelength pairs than are necessary for ascertaining the analyte. However, using a single scalar to represent the independent Mueller matrices per angle and wavelength demands considerably less computer memory.

Incorporating noise into the theoretical data and substituting in Equations (45) and (46) provides a way to simulate experimental uncertainties. One way to do this is to add a random component to the input variables of RETRO. For example, slightly varying the rough surface geometry (mean square height and slope) in a random manner simulates a scanning incident beam irradiating areas sample-to-sample.

#### 5.2.4 DECIDE2: A Detection Optimization Algorithm.

Program DECIDE2 computes and analyzes backscatter Mueller matrices every time it calls its subroutine RETRO. These data are used to better distinguish between background (base) and target (analyte) materials. In performing its intended function, DECIDE2 determines which Mueller matrix elements are of use at wavelengths and incident angles susceptible to the analyte.

DECIDE2 is an alternative to using the DISPLAY plotting package for graphical discrimination analyses. DECIDE2 locates primary resonant wavelengths for each material. It then locates the beam wavelengths at which the difference in the imaginary part of refractive index between target and background are maximum. At each of these wavelengths, DECIDE2 computes Mueller matrices for both materials as a function of incident angle. Immediately following this computation, each pair of corresponding Mueller matrices is separately analyzed (Equations 37 and 38).

The program DECIDE2 identifies the combination of these wavelengths and incident angles that result in a probable discrimination between the two unlike materials. These angle/wavelength pairs are slightly varied and reexamined. If there is an increase in discrimination characteristics between varied angles and wavelengths, then the program stores those new parameters. A new variation in angle and wavelength about these values are interrogated next, and so on. This 'seeking' program iterates the interrogation process until no further increase in discrimination has been detected. Once the program has located the optimum angle and wavelength, the computer proceeds with its analysis of 121 more Mueller matrices for angles of incidence and wavelengths near other initial optimum pairs. These 121 Mueller matrices, with analysis results, are written in a file of format readable to DISPLAY.

## 6. DISCUSSION OF FUTURE WORK

An alternate digital method for data acquisition, and a neural network interface to the analog detection output module are presented in this section. The new digital processing methods we are now exploring should result, if successful, in a turn-key data acquisition unit with on-board functions that filter specific frequencies in the scattered light intensity much like the lock-in electronics of the analog data acquisition.

### 6.1 Digital Data Acquisition and Signal Processing of the Scattergram.

We begin this section by summarizing the current method, which uses separate lock-in amplifiers, for determining the normalized Mueller matrix elements at a given wavelength and scattering angle. Let us assume that the driving amplitudes on the two photoelastic modulators have been properly set for the wavelength in use, and that the angles and orientations of all the optical elements have been correctly adjusted. Then Equation 11 shows that the output from the detector can be represented as the sum of an infinite number of discrete frequencies, namely the sums and differences of all integral multiples of the two modulator frequencies. The amplitude of each frequency component is given by the product of one or two Bessel functions of integer order (which fortunately tend toward zero as the order increases) and a factor that is one of eight Mueller matrix elements. The dc component of the detector output is proportional to a ninth Mueller matrix element, the  $\psi_{11}$  element. When the detector system's gain is actively servo-controlled to keep the dc output at a constant level, the ac components are also bounded so that their amplitudes are effectively proportional to the normalized Mueller matrix elements.

The same set of frequencies and Bessel function factors comprise the detector signal in each of the four experimental configurations: Case A,B,C, and D (see Section 4.3). All that differs among the configurations is the identity of the eight normalized Mueller matrix elements that help determine the amplitude of those frequencies.  $\psi_{11}$  is proportional to the detector dc level in each configuration.

Measurement of a normalized Mueller matrix element then is equivalent to the measurement of the amplitude of the corresponding frequency component in the detector signal. That is done initially with eight separate lock-in amplifiers, one for each frequency. Eight sinusoidal reference signals - each of which is at the same frequency as and synchronized with one of the eight desired frequencies in the detector signal - are produced by appropriate analog multiplications and filtering among sine waves derived from the reference outputs of the two PEM power supplies. The detector signal is split and sent to each lock-in amplifier board where it is first filtered through a narrow passband filter to reject most of the power except that near the desired frequency for which the particular board is designed. When the enhanced signal is finally multiplied against its corresponding reference frequency by the lock-in amplifier, the dc component of the resulting waveform is a measure of that frequency's amplitude in the detector signal and so is a measure of the corresponding (depending on experimental configuration) normalized Mueller matrix element.

Note that Equations 12, 14b, 16b, and 18b for the Mueller matrix elements are relationships among optical quantities - retardation (radians) of the two modulators and intensity ( $\text{Watt cm}^{-2}$ ) incident on the detector. The voltages presented to the lock-in amplifiers' inputs are, in a sense, representations of those optical quantities. But between the optics and the lock-ins lie a great many electronic components that transform, amplify, and filter signals along the way. As a result, an expression for the Mueller matrix elements analogous to, say Equation 12, but in terms of the voltages at the lock-ins would have to include factors for the gains and phase shifts (both frequency dependent) introduced by the train of electronics. Tracking all this would be impractical; instead, on each lock-in board is included a phase

shifter to shift the reference frequency relative to the signal, and a final gain control. These are adjusted and set on each board during the calibration procedure in which measurements are made for optical standards (polarizers and waveplates) whose Mueller matrix elements are known. After calibration, a reference frequency and its corresponding detector signal component will always be in phase or 180 degrees out of phase at the lock-in, depending on the sign of the corresponding Mueller matrix element.

The lock-in amplifiers will operate with a short time constant, probably within a few tenths of a second, depending on the amount of detector noise present. In other words, the (dc) output of the lock-in at any instant depends only on the input voltages over the previous few tenth seconds. It has occurred to us that if we can digitize the detector output waveform over that period of time with adequate resolution, along with waveforms representing the simultaneous modulator retardations, we ought to be able to calculate the same information that the lock-ins give and so eliminate most of the experiment's data acquisition electronics.

Three separate techniques for computing Mueller matrix elements from digitized data have suggested themselves already, and appear plausible to warrant serious investigation. We have not yet worked out all the details for any approach, but the concepts involved will be sketched out below.

The most obvious approach is to measure a fast Fourier transform (FFT) on a data stream sampled from the detector and note the amplitudes at the eight frequencies of interest. A calibration relating each Fourier amplitude to the corresponding normalized Mueller matrix amplitude would need to be performed, but in principle a simple power spectrum of the detector output will yield the magnitudes of the Mueller matrix elements. A greater effort is required to decide the signs of the elements: the FFT must compute the phase of each signal component as well as its amplitude, and the complex FFT must also be performed on simultaneously sampled sine waves synchronized with the two polarization modulators. From those three phases, with perhaps a phase correction determined in the calibration procedure, the sign of the Mueller matrix element can be worked out.

At least two ways of implementing the FFT approach are feasible. A sophisticated multichannel waveform analyser, such as the Analog 6100 or LeCroy 9424 in our laboratory, can acquire the waveforms and measure the FFT's rapidly (Figure 19). The resulting amplitudes and phases would be transferred to a PC for the final arithmetic and display and/or storage of the Mueller matrix elements. Alternately, the entire process could be carried out with a real time microcomputer or PC, using an A/D board with at least three input channels to acquire the waveforms, and software including a FFT routine to analyze them and extract the Mueller matrix elements.

A second approach is to let the computer emulate the system now in use by carrying out numerically the same multiplications and filtering that the analog electronics perform. Here, it would be more convenient to synchronize a pair of reference sawtooth waveforms (rather than sine waves) with the polarization modulators, so that a sampled voltage represented the instantaneous phase (rather than amplitude) of the retardation of its modulator. Then the simultaneous phases of the remaining six reference frequencies could be quickly calculated from sums and differences of the two sampled phases.

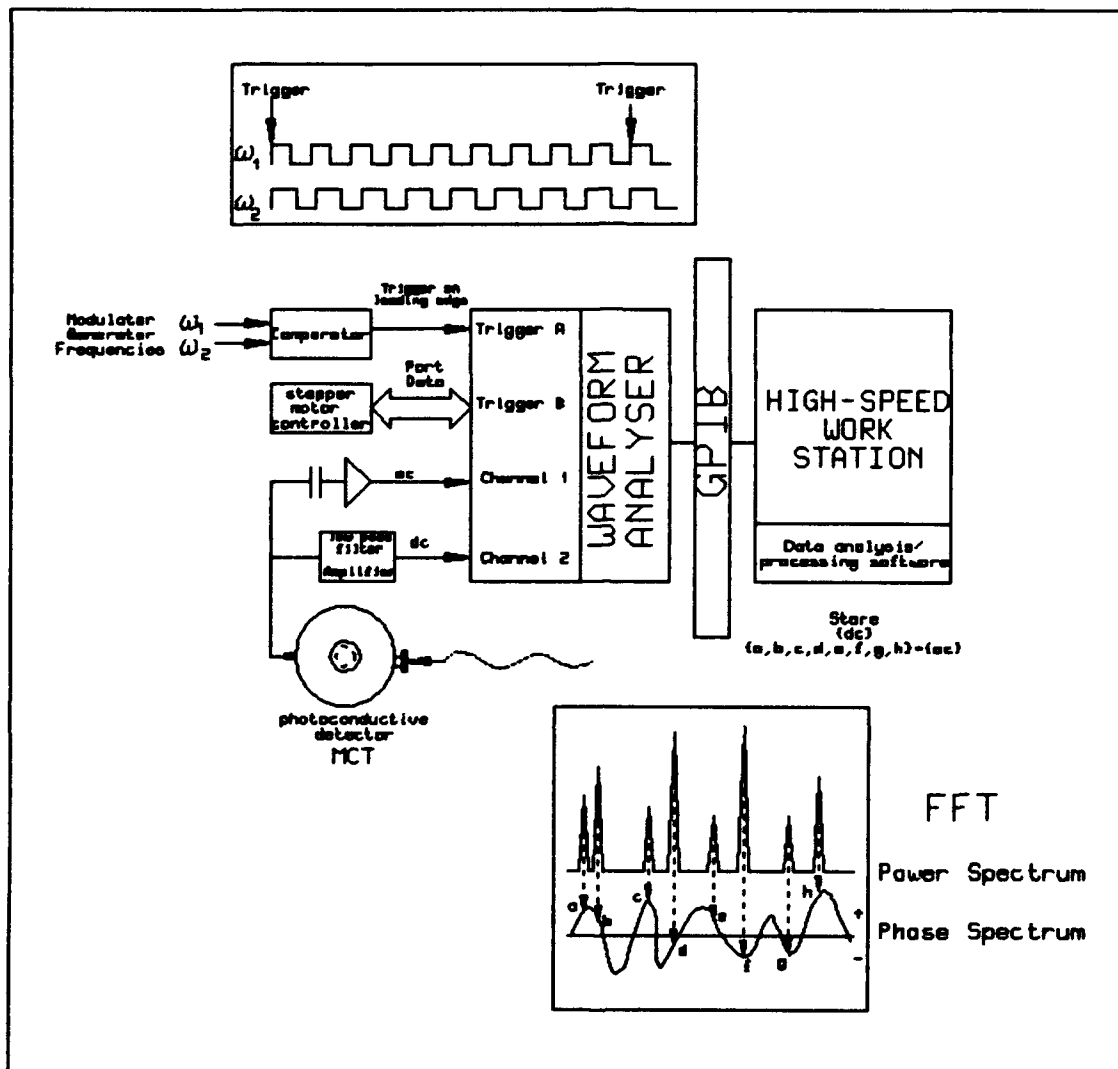


Figure 19. Mueller matrix digital acquisition using a standard Analogic 6100/650 or LeCroy 9424 waveform analyser. This instrument computes Fourier intensity and phase spectra of the MCT-detected scattergram. Acquisition of Mueller elements are triggered when the modulator reference frequencies are aligned as shown at the top of the figure. They are correlated to peak Fourier intensities a, b, c, d, e, f, g, h and a dc value that make up the scattergram waveform. The nine Mueller elements are transferred to computer memory via an ANSI command from the CPU routed through the stepper motor controller, used to control the experimental operation.

We envision something like the following sequence of operations. Three voltages are read in and scaled via a A/D board, representing, at the same instant, the phase of transmitter modulator ( $\omega_1$  radians), the phase of receiver modulator ( $\omega_2$  radians), and the detector output. From the two phases we form the instantaneous phases of the other six references:  $2\omega_1$ ,  $2\omega_2$ ,  $\omega_1 - \omega_2$ ,  $2\omega_1 - \omega_2$ ,  $\omega_1 - 2\omega_2$ , and  $2\omega_1 - 2\omega_2$ . (In some cases sums instead of differences may be chosen.) At this point compensating phase shifts, previously determined in a calibration procedure, may be applied to the eight phases. The cosine of each net phase is then calculated, giving eight numbers that represent the instantaneous values of the eight reference frequency voltages that (in the present analog system) would be found on the reference inputs of the lock-in amplifiers. Note that the amplitude of a reference frequency is not a relevant quantity, so long as it's constant, and is here taken to be unity.

When each of the cosine terms is multiplied by the detector output, there results eight numbers that represent the instantaneous outputs of the present eight lock-in amplifiers (without low pass output filtering). Recall that each Mueller matrix element is proportional to the average level (dc component) of its lock-in output. Electronically that level is determined with a low pass filter smoothing the output over some period of time ( $\approx$  time constant). We can accomplish the same thing numerically by repeating the measurements just described many times over the same time period and taking averages. Thus, in eight computer memory locations we would accumulate (add) the eight effective lock-in amplifier outputs calculated each time a triplet of data points ( $\omega_1$ ,  $\omega_2$ , detector) were read in. After enough readings (thousands?) are gathered over a sufficient length of time (.5 sec?), each of the eight accumulated numbers would be divided by the total number of readings and scaled by a fixed factor - previously determined by calibration - to give the value of the corresponding Mueller matrix element.

In the third approach the frequency content of the detected signal isn't considered at all. Instead, our starting point is Equation 12, 14b, 16b, or 18b; an exact closed equation relating the detected intensity to the retardation on the two polarization modulators and the Mueller matrix elements. Suppose, as in the last approach (lock-in emulation), we read in a triplet of values representing  $\omega_1$ ,  $\omega_2$  and the detector signal. Then, taking  $\psi_{11} = 1$ , we can evaluate all the quantities of the above equations except the eight Mueller matrix elements, giving one equation with eight unknowns. Reading in seven more triplets of values will yield a total of eight equations in eight unknowns, which can then be solved for the Mueller matrix elements by standard techniques, such as an inverse matrix calculation. This process might be repeated often in a very short time and averages taken to reduce the influence of experimental noise and the occasional (?) ill-conditioned data set.

We wish to investigate soon whether one or more of these three data processing techniques - or perhaps other techniques not yet thought of - can replace the rack of analog electronics now used. All three approaches should be easy to implement on a microvax or PC system, and should be tested using synthetic and/or real data. There are many questions and problems to be considered, such as the density and total number of data points required in each approach, how to reconcile the need for simultaneous data triplets with the sequential nature of multiplexer data acquisition, and the stability of the solutions obtained in the face of experimental noise.

## 6.2 Rapid Laser Switching Between Resonance-Reference and Resonance-Resonance Beam Wavelengths.

A latter objective of this program is field evaluation in near real time of Mueller matrix elements measured in succession between rapidly switching beams irradiating surfaces down-range to kilometer distances. To accomplish these more distant and more rapid measurements, the ellipsometer system will be expanded to incorporate more powerful lasers and a larger receiver collection aperture.

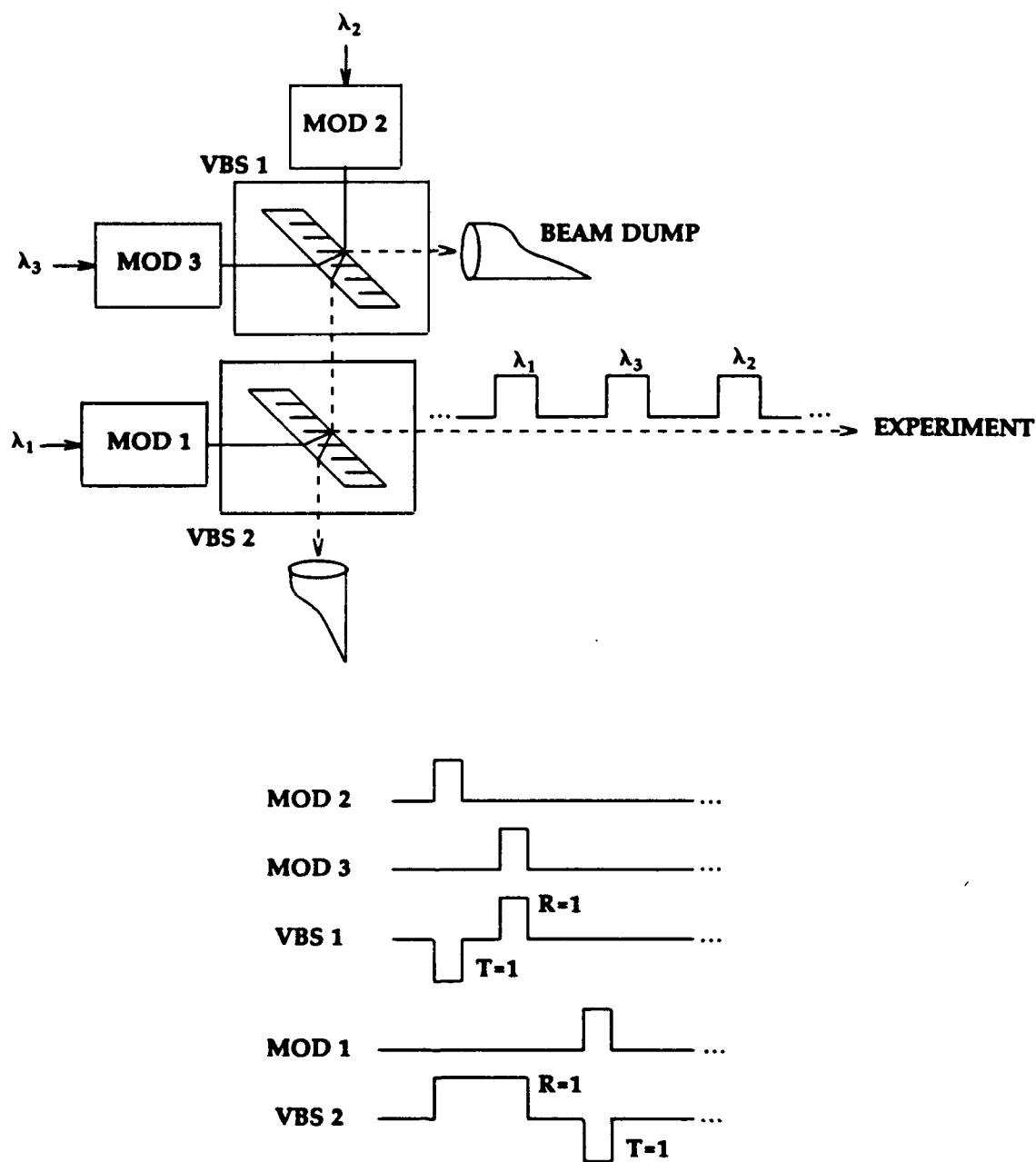
The transmitter of this future system was eluded to in the previous Figure 4c. Let us return to that same type of configuration, but for the sake of simplicity consider here a three-wavelength switching transmitter system.

The variable beam splitters (VBS1-2) produce full transmission and reflection modulation from  $< .05$  to  $> .95 R$  between 9 and  $12.5 \mu m$ , via a piezo electric interface control module. (For n-laser wavelength pulsing, n-1 VBS optics are required.) The VBS modulators work on a principle of Frustrated Total Internal Reflection. (The technique is proprietary to the optics manufacturer.) Two VBS optical systems are now being custom designed for these ellipsometers by the Kentek Corporation, Laser Tools Division.

Amplitude modulation and triggering of the three incident cw beams is accomplished internally in the laser's power supplies and exciter circuits. Amplitude modulators MOD1-3 designate switching access via TTL logic signals to the power supplies, as shown in Figure 20.

Our concern with this transmitter is the purity of polarization modulation between pulses. The pulsing is adjustable from 10 to 100 milliseconds or greater. Another concern of the frequency agile ellipsometer systems is what tolerance the modulators can withstand on consistency of periodic phase retardation adjustments between pulses of unlike wavelength. Air-cooled ZnSe can apparently operate under a maximum 100 watt beam intensity without significant damage to its anti-reflection coating. However, maintaining a constant retardation ( $\delta_0$  in Equation 9) in the PEM's between beam pulses is a stringent constraint placed on the resonant compression and relaxation induced on the ZnSe crystals.

We also have future plans to utilize the dead time between beam pulses in an integrated pseudo active emissions fusion sensor concept, where chemical vapor contamination and liquids on a surface are detected spectroscopically in thermoluminescence produced from heating by the beam, and subsequent release of mid IR Planck emissions.<sup>36</sup> The thermoluminescence sensor component of the system would consist of a solid-state interferometer based on the same PEM technology incorporated in the ellipsometer systems. This will be discussed in a later report.



**Figure 20.** Rapid 3-laser switching based on variable beam splitter (VBS) technology. MOD 1,2 and 3 are amplitude modulators of the four incident beams, and VBS 1,2 are the electronically controlled transmitting/reflecting (T/R) beam splitters. Pulse and triggering sequences generating the train of alternating wavelengths  $\lambda_2:\lambda_3:\lambda_1$  are shown in the bottom half of the figure. Pulse gating and beam durations can be varied by interface to the piezo electric circuitry.

### 6.3 Neural Network Computing of the Mueller Elements For Standoff Analyte Detections.

Work has begun on applying a neural network to the analog APSD detector outputs for purposes of contaminant decision making and density mappings. As the name implies, neural network systems intend to emulate the brain's parallel processing ability by activating a set of impulses (in this case, real-time information from 16 independent channels of Mueller elements analog outputs from the sensor), pass it along weighted interconnecting nodes (the neurons, weights via a valid theoretical model) that transform these data to a system of hidden layers and other nodes, where new transforms operate on these impulses to produce an output layer. The pattern of information from the network's final output layer (back- or forward-propagated) has interpretation that may correlate to a detection event or non-event.

All networks we are considering are constructed from interconnected nodes, each of which forms a weighted sum of the Mueller matrix element inputs to the node, and adds a threshold value to the weighted sum. The value of this sum plus the threshold is passed through a nonlinearity, and the value of the non linear 'impulse' function is the output of the node. The inputs to each node are a combination of outputs from other nodes and primary Mueller element inputs to the network. The threshold of each node can be viewed as a unit weight for an input.<sup>37</sup> Generally, the connection weights and thresholds can be adapted using iterative procedures to make the network produce a desired output when a particular input is presented. Many of these network concepts have been demonstrated to work well when the input data is noisy.

The detection network application is schematized in the following Figure 21. We refer to Lippmann's paper<sup>38</sup> and references therein for a description of various neural network architectures.

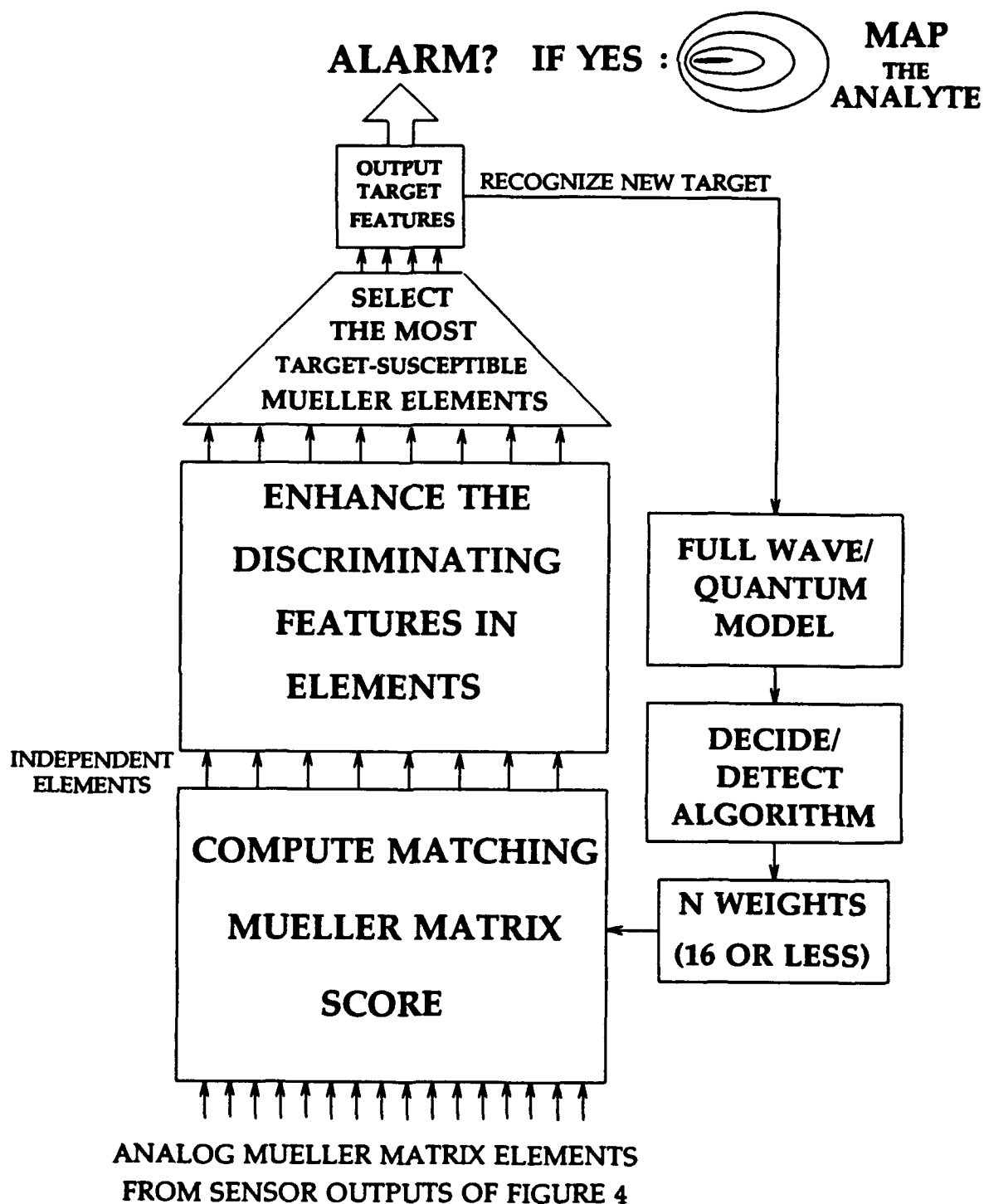


Figure 21. A neural network application for accessing elements of the Mueller matrix in real-time operation of the ellipsometer sensor. Inputs from theoretical and statistical models weigh all discriminating analyte features from the full 16-element matrix field. The successful network architecture will adapt to new analytes (contaminants) from the scattering sample, weighing elements according to their intramolecular phase signatures. The goal of the network is to alarm against the analyte (if present), and map its mass density.

Determining and optimizing a particular architecture that can be best implemented in construction of a Mueller matrix hardware network to serve as a contaminant classifier is a topic of future investigations. Some candidate nets are Hopfield-, Hamming-, Grossberg-, and Kohonen-like architectures. For the APSD sensor-specific network, the best choice of the number of nodes and hidden layers is an important first step in network development.

Neural networks can be used for content-addressable memory, vector quantization, data clustering and pattern recognition. The remote detection application pursued here requires a net that performs the last two of these functions. Two networks that form clusters are the Carpenter/Grossberg classifier and Kohonen self-organizing feature mapper.

The first network architecture we have investigated is a Rumelhart-McClelland single layer perceptron structure using three nodes. The network structure is given as follows.

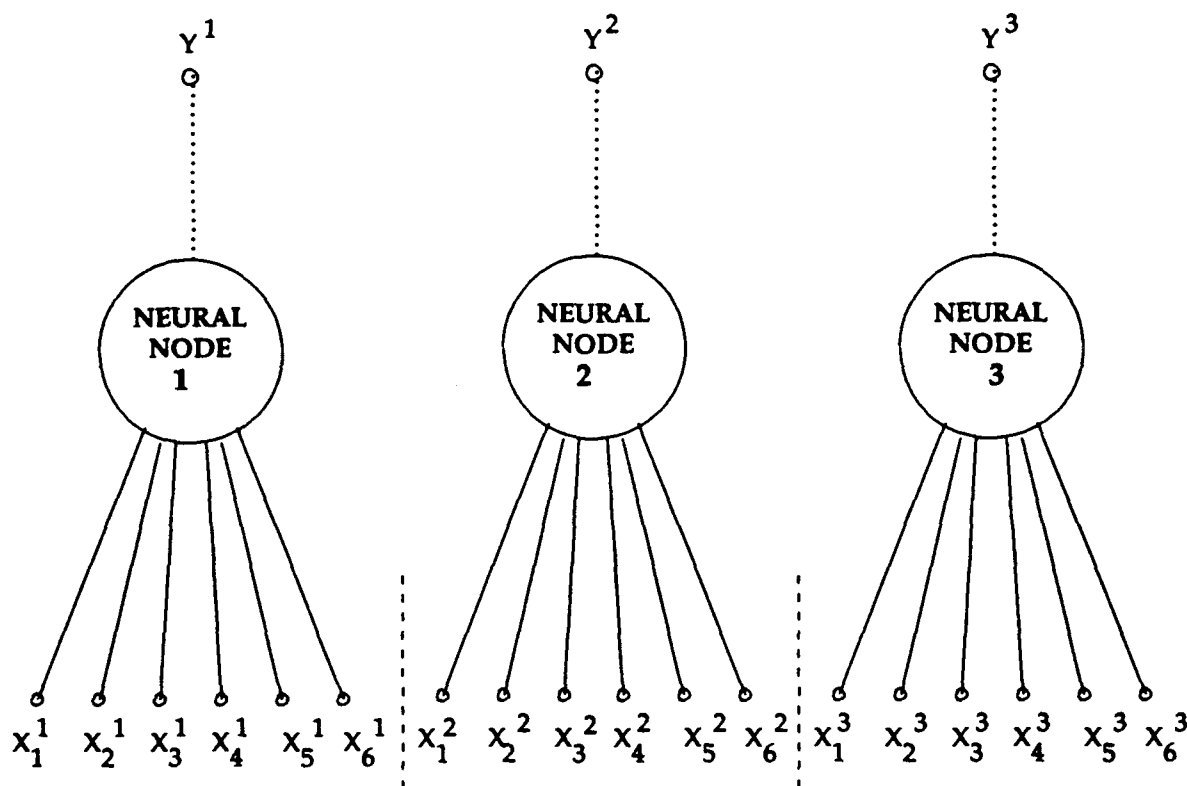


Figure 22. The architecture of a neural network that implements a detection algorithm for the near real-time identification of SF96, DMMP, and/or DIMP.

Each node of the neural network determines on which side of a hyperplane an input Mueller matrix lies. The Mueller matrix data from the subroutine RETRO indicates (with added noise) scattering from various coating materials and unwetted surfaces, separated by a hyperplane. The partitioning of the Mueller elements into classes is done at backscattering angles and wavelength combinations most susceptible to the analyte. Those parameters are obtained from the programs DETECT and DECIDE. Judicious initial choices for the connection weights and thresholds into the network will yield the correct output at the onset of presentation of training data. Several sets (10-30 per backscatter angle-wavelength pair) of noisy data

must be taken in order to obtain the standard deviation values for the Mueller elements. These are required for selecting a good initial choice of connection weights and node thresholds.

This neural network model is coded in Fortran 77 and now running on a CRAY2 computer facility. When implemented, it is capable of detecting SF96, DMMP, and DIMP. (Since the three analytes can be identified using three different combinations of backscatter angle and wavelength, the network requires three independent nodes.) The parameters in Figure 22 are defined as follows:

$$Y^i = f_h \left( \sum_j w_j^i X_j^i + \theta^i \right) \quad (47)$$

For the output node 1,  $Y^1$  is high (=1) if SF96 is present, and low (=0) otherwise. The output of node 2(3),  $Y^2(Y^3)$  is high if DMMP(DMMP) is present, and low otherwise. The values  $w_j^i$  are connection weights from inputs  $X_j^i$  to node  $i$ ,  $\theta^i$  is the threshold at node  $i$ , and  $f_h$  is a hard limiting nonlinearity ( $f_h = 0$  if the argument is negative, and 1 if positive.) The absolute value of the  $j^{th}$  Mueller element at the  $i^{th}$  angle-wavelength combination is the network input:

$$X_j^i = |f_{mn}(\lambda_i, \theta_i)| \quad (48)$$

where  $j=1-mn=11$ ,  $j=2-mn=12$ ,  $j=3-mn=22$ ,  $j=4-mn=33$ ,  $j=5-mn=34$ , and  $j=6-mn=44$ . To be consistent with previous notations,  $i=1,2,3$  designates detection of SF96, DMMP, and DIMP respectively.

The connection weights  $w_j^i$  and thresholds  $\theta^i$  can be adapted using a perceptron convergence procedure<sup>38</sup>. The iterations required for  $w$  and  $\theta$  to converge can be reduced by using the following initial values:

$$\theta^i = -\frac{1}{n} \sum_j \frac{X_{b_j}^i + X_{t_j}^i}{\sigma_{b_j}^i} \quad (49a)$$

$$w_j^i = \frac{1}{\sigma_{b_j}^i} u(X_{t_j}^i - X_{b_j}^i), \quad (49b)$$

where  $u(\cdot)$  is a unit step function,  $\sigma_{b_j}^i$  is the standard deviation of Mueller element  $j$  for the unwetted (bare) surface, and inputs  $X_{b(t)}^i$  are the absolute values of the Mueller elements  $j$  for unwetted (wetted) surfaces at the susceptible  $\lambda$ - $\theta$  values. Even though these initial  $w$  and  $\theta$  choices near eliminate a need for training sessions, several sets of data are required to obtain accurate values for  $\sigma_{b_j}^i$ .

#### **6.4 Initial Experimental and Theoretical Mueller Matrix Data Bases.**

A thorough data base generated through the laboratory ellipsometer instruments and a successful model, to interpret these data, will determine the sensitivities and limitations of this technology toward solution of a particular detection problem. It will guide us in the selection of Mueller elements (as functions of backscattering angle, wavelength) that can identify an analyte or class of compounds alone, spread across a background surface, or disseminated as an aerosol. What can be done to filter in and enhance (optically or by mathematical algorithms) information by the analyte? That will be answered after a full and reproducible data base is produced, so that we can better understand the principles of polarized IR beam - surface interactions. Once this technology is understood, and if it can be proven feasible, prototype sensors will be designed and applied toward a specific problem, like detecting an agent wetting soil, a biological impurity in a specimen, an aerosol pollutant, an oxide growth on a semiconductor surface, and so on. We present here the plan of an initial data base where the goals are verification of model calculations and rapid detection of classes of analyte compounds in situ.

##### **6.4.1 Metals and Insulators of Known IR Refractive Index and Surface Geometry: Validation of the Full Wave Model.**

In Sections 5.2.1-5.2.4 and 6.3 we had discussed the Full Wave electromagnetic scattering theory, and how it can be applied toward: (1) initializing the ellipsometer sensor for maximum probability of a successful analyte detection, (2) simulation of the entire experimental operation, and (3) fabricating a neural network discriminator. The theory, of course, would have to be experimentally verified before these applications can be implemented. We have set out to prove (or disprove) its predictions via a control set of scattering experiments from aluminum, graphite, and other surfaces of known optical constants over the IR, and known geometry (surface slopes and heights). (The surfaces are etched or sand blasted, and surface-profiled in 3-dimensions by an interference-type instrument.) The general experimental procedure will involve scanning the entire range of backscattering angles over many of the laser transitions, comparing results with model data such as those presented in Figures 16-18. The University of Nebraska group (E. Bahar) is now expanding the Full Wave to include scattering from multi-layered structures and non-isotropic scatterers. When the new versions of RETRO code are written, predicted scattering signatures will be compared with data measured by this ellipsometer sensor.

##### **6.4.2 Biological and Controlled Substance Simulants.**

The Mueller matrix ellipsometer produces a full optical description of the scatterer by its response to a continuous span of linear and elliptical polarization states over selected frequencies of the irradiating beams. The emphasis here is complete characterization of linear scattering processes. Spectroradiometers that measure absorption bands in collected radiance<sup>36</sup> cannot resolve molecularly similar compounds with overlapping extinction bands - compounds that may have entirely different toxicity!! With phase-sensitive scattering, we would like to test whether isomers (molecules of identical molecular weight but different group symmetry) can be distinguished, for instance, through their dichroism signatures. Many biological compounds contain segments that are chiral, thus the ability to distinguish between chirality has applications of biological and contraband detection. The first sets of experiments to test this uniqueness assertion will be conducted on the biological and controlled substance simulants listed in Table 5. In it are compiled the scatterer's molecular formula, its strongest resonant absorption over the IR, and the nearest ellipsometer laser energy to that absorption center

frequency. Typically, three of the four lasers are tuned to a distinct absorption (analytical) band and the fourth is off-tuned to a region of non-resonance in the analyte (reference). The ratios of analytical to reference Mueller elements found most susceptible to the contaminant's optical activity are then sought for making a detection decision.

#### **6.4.3 Chemical Agent Simulants.**

The experiments with liquid simulants of chemical agent are trained toward their strong absorption bands, in some agents the P=O, C-O and P-O-C stretching vibrations are most important. In these experiments, Mueller elements of the bare surface are measured. Next, the analyte is ejected (via an aerosol deposition, to simulate an actual agent attack) in low concentration, and Mueller elements remeasured. This continues on to higher concentrations and element re-measurements. A pattern is established in the Mueller elements with concentration of the analyte. Table 5 lists the absorption properties of four common agent simulants: DMMP, DIMP, SF96, and DEP.

#### **6.4.4 Interferents.**

Interferents are all scatterers other than the analyte. The analyte scattering signal is usually a small superposition on the interferent scattering signal, and must be observed in the differential resonance/non resonance Mueller elements for successful detections. Fortunately, terrains (a sum of quartz, kaolinite, illite, montmorillonite and other minerals) are broadband absorbers of IR radiation and the analyte compounds have sharp extinction frequencies. Other interferents such as diesel soots, fog oils,<sup>21</sup> possess their own absorption moieties. Thus, the susceptible Mueller elements can single out the analyte on a surface at the (very narrow) laser line by adjusting the ellipsometer beams to the analyte's center extinction frequencies, ratio these elements to those measured at a reference laser transition, then subtract this result to the bare surface Mueller elements. In Table 5, the common minerals found in soil are listed, all are broadband IR absorbers.

**Table 5.** Candidates for characterization through select IR Mueller matrix signatures. A data bank of Mueller elements is established per absorption wavelength (plus a minimum of one reference laser wavelength) over all back-scattering angles, and organized in a computer file similar to this table's format. The major absorption bands by each material are listed, as are their nearest matching laser line over the ellipsometer's 4-laser bandwidth (see Appendix I).

Scatterer	Formula	Major Vibration ( $\text{cm}^{-1}$ )	Nearest Laser Line to the Scatterer's Vibrational Frequency				Comment
			Transition	Freq. ( $\text{cm}^{-1}$ )	Band	Type Laser	
D-(-)-Arabinose	$\text{C}_5\text{H}_{10}\text{O}_5$	842.7	P(28)	842.79	$00^01 \rightarrow 10^00$	$\text{C}^{14}\text{O}_2^{16}$	Chiral Organics (Biological Simulant Analytes)
		892.5	P(26)	891.57	$00^01 \rightarrow 10^00$	$\text{C}^{13}\text{O}_2^{16}$	
			R(36)	892.04	$00^01 \rightarrow 10^00$	$\text{C}^{14}\text{O}_2^{16}$	
		1000.6	P(64)	1000.82	$00^01 \rightarrow 02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(20)	1000.65	$00^01 \rightarrow 02^00$	$\text{C}^{13}\text{O}_2^{16}$	
			R(26)	1000.95	$00^01 \rightarrow 02^00$	$\text{C}^{14}\text{O}_2^{16}$	
		1052.4	P(14)	1052.20	$00^01 \rightarrow 02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(40)	1052.26	$00^01 \rightarrow 02^00$	$\text{C}^{12}\text{O}_2^{18}$	
DL-Alanine Monohydrate	$\text{C}_3\text{H}_7\text{NO}_2$	852.0	P(16)	853.2	$00^01 \rightarrow 10^00$	$\text{C}^{14}\text{O}_2^{16}$	
D-Alanine	$\text{C}_3\text{H}_7\text{NO}_2$	850.6	P(18)	851.50	$00^01 \rightarrow 10^00$	$\text{C}^{14}\text{O}_2^{16}$	
DL-Aspartic Acid	$\text{C}_4\text{H}_7\text{NO}_4$	1073.1	R(12)	1073.28	$00^01 \rightarrow 02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(14)	1073.58	$00^01 \rightarrow 02^00$	$\text{C}^{12}\text{O}_2^{18}$	
L-Aspartic Acid	$\text{C}_4\text{H}_7\text{NO}_4$	1045.9	P(22)	1045.02	$00^01 \rightarrow 02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(48)	1045.08	$00^01 \rightarrow 02^00$	$\text{C}^{12}\text{O}_2^{18}$	
(-)-Atropine Sulfate Monohydrate	$\text{C}_{17}\text{H}_{23}\text{NO}_3$	967.3	R(8)	967.71	$00^01 \rightarrow 10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(18)	967.45	$00^01 \rightarrow 02^00$	$\text{C}^{14}\text{O}_2^{16}$	
		1023.8	P(44)	1023.19	$00^01 \rightarrow 02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			R(6)	1022.93	$00^01 \rightarrow 02^00$	$\text{C}^{13}\text{O}_2^{16}$	
		1073.9	R(12)	1073.28	$00^01 \rightarrow 02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(14)	1073.57	$00^01 \rightarrow 02^00$	$\text{C}^{12}\text{O}_2^{18}$	

Table 5 - continued.

Scatterer	Formula	Major Vibration(s) ( $\text{cm}^{-1}$ )	Nearest Laser Line to the Scatterer's Vibrational Frequency				Comment
			Transition	Freq. ( $\text{cm}^{-1}$ )	Band	Type Laser	
(1R)-(+)-Camphor	$\text{C}_{10}\text{H}_{16}\text{O}$	853.5	P(16)	853.2	$00^01-10^00$	$\text{C}^{14}\text{O}_2^{16}$	+ VCD (strong)
		935.0	P(30)	934.89	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(38)	935.89	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{18}$	
			R(30)	935.14	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	- VCD(strong)
		1045.3	P(22)	1045.02	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(48)	1045.08	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
(1S)-(-)-Camphor	$\text{C}_{10}\text{H}_{16}\text{O}$	934.9	P(30)	934.89	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(38)	935.89	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{18}$	
			R(30)	935.14	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	
		1045.3	P(22)	1045.02	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(48)	1045.08	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
( $\pm$ )-Camphor	$\text{C}_{10}\text{H}_{16}\text{O}$	1045.1	P(22)	1045.02	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(48)	1045.08	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
L-Cysteine	$\text{C}_3\text{H}_7\text{NO}_2\text{S}$	867.2	R(4)	869.96	$00^01-10^00$	$\text{C}^{14}\text{O}_2^{16}$	
		1064.8	R(0)	1064.51	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(26)	1064.13	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
D-(-)-Fructose	$\text{C}_6\text{H}_{12}\text{O}_6$	978.2	R(24)	978.47	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			R(18)	978.89	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{18}$	
			P(8)	976.21	$00^01-02^00$	$\text{C}^{14}\text{O}_2^{16}$	
		1079.6	R(22)	1079.85	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(6)	1079.49	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
D-Glucose Anhydrous	$\text{C}_6\text{H}_{12}\text{O}_6$	838.7	P(32)	839.20	$00^01-10^00$	$\text{C}^{14}\text{O}_2^{16}$	
		995.7	R(56)	995.07	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(26)	994.99	$00^01-02^00$	$\text{C}^{13}\text{O}_2^{16}$	
			R(16)	994.82	$00^01-02^00$	$\text{C}^{14}\text{O}_2^{16}$	
		1024.1	P(44)	1023.19	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			R(8)	1024.37	$00^01-02^00$	$\text{C}^{13}\text{O}_2^{16}$	
L-Histidine	$\text{C}_6\text{H}_7\text{N}_3\text{O}_2$	924.5	P(40)	924.97	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			R(14)	924.53	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	
Glycine	$\text{C}_2\text{H}_5\text{NO}_2$	893	P(26)	891.57	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	
		912	P(54)	907.77	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
		936	P(28)	936.80	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(38)	935.89	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{18}$	
			R(32)	936.37	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	
		1033	P(34)	1033.48	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(58)	1035.70	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	

Table 5 - continued.

Scatterer	Formula	Major Vibration(s) ( $\text{cm}^{-1}$ )	Nearest Laser Line to the Scatterer's Vibrational Frequency				Comment
			Transition	Freq. ( $\text{cm}^{-1}$ )	Band	Type Laser	
S-(-)-Limonene	$\text{C}_{10}\text{H}_{16}$	887.1	P(30)	887.92	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	
			R(28)	886.93	$00^01-10^00$	$\text{C}^{14}\text{O}_2^{16}$	+ VCD (strong)
		914.2	P(50)	914.42	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	+ VCD (weak)
			R(4)	917.25	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	
		1051.4	P(16)	1050.44	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	- VCD (very weak)
			P(42)	1050.49	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
D-Mannose	$\text{C}_6\text{H}_{12}\text{O}_6$	969.9	R(10)	969.14	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			R(4)	970.33	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{18}$	
			P(16)	969.26	$00^01-02^00$	$\text{C}^{14}\text{O}_2^{16}$	
		1040.0	P(28)	1039.37	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(54)	1039.50	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
			R(32)	1039.38	$00^01-02^00$	$\text{C}^{13}\text{O}_2^{16}$	
D-(-)-Ribose	$\text{C}_5\text{H}_{10}\text{O}_5$	911.7	P(54)	907.77	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
		959.0	P(2)	959.39	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(10)	959.71	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{18}$	
			P(26)	959.90	$00^01-02^00$	$\text{C}^{14}\text{O}_2^{16}$	
			P(30)	1037.43	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
		1037.0	P(56)	1037.61	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
			R(28)	1037.17	$00^01-02^00$	$\text{C}^{13}\text{O}_2^{16}$	
L-(-)-Sorbitol	$\text{C}_6\text{H}_{12}\text{O}_6$	820.2	P(48)	824.17	$00^01-10^00$	$\text{C}^{14}\text{O}_2^{16}$	
		883.1	P(34)	884.18	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	
			R(22)	882.91	$00^01-10^00$	$\text{C}^{14}\text{O}_2^{16}$	
		991.8	R(48)	991.57	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			R(44)	991.27	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	
			P(30)	991.07	$00^01-02^00$	$\text{C}^{13}\text{O}_2^{16}$	
			R(10)	990.79	$00^01-02^00$	$\text{C}^{14}\text{O}_2^{16}$	
		1047.9	P(18)	1048.66	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(44)	1048.71	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
L-Serine	$\text{C}_3\text{H}_7\text{NO}_3$	1013.1	P(52)	1014.52	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
L-Threonine	$\text{C}_4\text{H}_9\text{NO}_3$	936.2	P(28)	936.80	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(38)	935.89	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{18}$	
			R(32)	936.37	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	
		1041.1	P(26)	1041.28	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(52)	1041.38	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
			R(36)	1041.48	$00^01-02^00$	$\text{C}^{13}\text{O}_2^{16}$	

Table 5 - continued.

Scatterer	Formula	Major Vibration(s) ( $\text{cm}^{-1}$ )	Nearest Laser Line to the Scatterer's Vibrational Frequency				Comment
			Transition	Freq. ( $\text{cm}^{-1}$ )	Band	Type Laser	
L-Tyrosine	$\text{C}_9\text{H}_{11}\text{NO}_3$	841.3	P(30)	841.00	$00^01-10^00$	$\text{C}^{14}\text{O}_2^{16}$	
DL-Tartaric Acid	$\text{C}_4\text{H}_6\text{O}_6$	1094.9	R(16)	1094.76	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
L-Tartaric Acid Hydrate	$\text{C}_4\text{H}_6\text{O}_6$	943.0	P(22)	942.38	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(30)	943.23	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{18}$	
			R(44)	943.34	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	
		1087.8	R(36)	1087.95	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			R(4)	1087.11	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
D-Tartaric Acid	$\text{C}_4\text{H}_6\text{O}_6$	943.3	P(22)	942.38	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(30)	943.23	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{18}$	
			R(44)	943.34	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	
		1087.8	R(36)	1087.95	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			R(4)	1087.11	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
D-(+)-Xylose	$\text{C}_5\text{H}_{10}\text{O}_5$	903.7	P(60)	903.21	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
		934.2	P(30)	934.89	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(38)	935.89	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{18}$	
			R(30)	935.14	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	
		1039.8	P(28)	1039.37	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(54)	1039.50	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
			R(32)	1039.38	$00^01-02^00$	$\text{C}^{13}\text{O}_2^{16}$	
D-(-)-Xylose	$\text{C}_5\text{H}_{10}\text{O}_5$	903.6	P(60)	903.21	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
		934.1	P(30)	934.89	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(38)	935.89	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{18}$	
			R(30)	935.14	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	
		1039.8	P(28)	1039.37	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(54)	1039.50	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{18}$	
			R(32)	1039.38	$00^01-02^00$	$\text{C}^{13}\text{O}_2^{16}$	
L-Tryptophan	$\text{C}_{11}\text{H}_{12}\text{N}_2\text{O}_2$	1005	P(60)	1005.48	$00^01-02^00$	$\text{C}^{12}\text{O}_2^{16}$	
			P(16)	1004.28	$00^01-02^00$	$\text{C}^{13}\text{O}_2^{16}$	
			R(34)	1005.31	$00^01-02^00$	$\text{C}^{14}\text{O}_2^{16}$	
		918.0	P(46)	918.72	$00^01-10^00$	$\text{C}^{12}\text{O}_2^{16}$	
			R(6)	918.74	$00^01-10^00$	$\text{C}^{13}\text{O}_2^{16}$	

Table 5 - continued.

Scatterer	Formula	Major Vibration(s) ( <i>cm</i> <sup>-1</sup> )	Nearest Laser Line to the Scatterer's Vibrational Frequency				Comment
			Transition	Freq. ( <i>cm</i> <sup>-1</sup> )	Band	Type Laser	
Chemical Agent Simulant Analytes							
Dimethyl- methyl- phosphonate (DMMP)	CH <sub>3</sub> PO(OCH <sub>3</sub> ) <sub>2</sub>	1049	P(18)	1048.66	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup>	C-O stretch
			P(44)	1048.71	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>18</sup>	
		1061	P(4)	1060.57	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup>	C-O stretch
			P(30)	1060.84	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>18</sup>	
		1072	R(10)	1071.88	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup>	C-O stretch
			P(14)	1073.58	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>18</sup>	
Diisopropyl- methyl- phosphonate (DIMP)	CH <sub>3</sub> PO(OCH(CH <sub>3</sub> ) <sub>2</sub> ) <sub>2</sub>	995	R(56)	995.07	00 <sup>0</sup> 1-10 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup>	P-O-C vib
			P(26)	994.99	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>13</sup> O <sub>2</sub> <sup>16</sup>	
			R(16)	994.82	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>14</sup> O <sub>2</sub> <sup>16</sup>	
		1014	P(52)	1014.52	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup>	(P-O-C) vib
Polydimethyl- siloxane (SF96)	[-Si(CH <sub>3</sub> ) <sub>2</sub> O-] <sub>n</sub>	1034	P(34)	1033.48	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup>	(Si-O-Si) vib
			P(58)	1035.70	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>18</sup>	
			R(24)	1034.83	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>13</sup> O <sub>2</sub> <sup>16</sup>	
		1092	R(46)	1092.96	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup>	(Si-O-Si) vib
			R(12)	1092.29	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>18</sup>	
Diethyl- Phthalate (DEP)	C <sub>12</sub> H <sub>14</sub> O <sub>4</sub>	1017.7	P(50)	1016.72	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup>	
		1073.6	R(12)	1073.28	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup>	
			P(14)	1073.58	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>18</sup>	
Controlled Substance Simulants (Analytes)							
Methyl- Benzoate	C <sub>9</sub> H <sub>8</sub> O <sub>2</sub>	1027.3	P(40)	1027.38	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup>	
			R(12)	1027.16	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>13</sup> O <sub>2</sub> <sup>16</sup>	
Methyl- Acetate	C <sub>3</sub> H <sub>6</sub> O <sub>2</sub>	1048.2	P(18)	1048.66	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup>	
			P(44)	1048.71	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>18</sup>	
Atropine							ibid

Table 5 - continued.

Scatterer	Formula	Major Vibration(s) ( <i>cm</i> <sup>-1</sup> )	Nearest Laser Line to the Scatterer's Vibrational Frequency				Comment	
			Transition	Freq. ( <i>cm</i> <sup>-1</sup> )	Band	Type Laser		
Scopolamine Hydrobromide hydrate	C <sub>17</sub> H <sub>21</sub> NO <sub>4</sub>	1046.9	P(20) P(46)	1046.85 1046.90	00 <sup>0</sup> 1-02 <sup>0</sup> 0 00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup> C <sup>12</sup> O <sub>2</sub> <sup>18</sup>		
Tropine	C <sub>8</sub> H <sub>15</sub> NO	1047.4	P(20) P(46)	1046.85 1046.90	00 <sup>0</sup> 1-02 <sup>0</sup> 0 00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup> C <sup>12</sup> O <sub>2</sub> <sup>18</sup>		
3-Azabicyclo- [3.2.2] nonane	C <sub>8</sub> H <sub>15</sub> N	1079.4	R(20) P(6)	1078.59 1079.49	00 <sup>0</sup> 1-02 <sup>0</sup> 0 00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup> C <sup>12</sup> O <sub>2</sub> <sup>18</sup>		
Benztropine Mesylate	C <sub>21</sub> H <sub>25</sub> NO	1050.1	P(16) P(42)	1050.44 1050.49	00 <sup>0</sup> 1-02 <sup>0</sup> 0 00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup> C <sup>12</sup> O <sub>2</sub> <sup>18</sup>		
Nipecotic Acid	C <sub>8</sub> H <sub>11</sub> NO <sub>2</sub>	1067.1	R(4) P(22)	1067.54 1067.36	00 <sup>0</sup> 1-02 <sup>0</sup> 0 00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup> C <sup>12</sup> O <sub>2</sub> <sup>18</sup>		
Piperidine	C <sub>5</sub> H <sub>11</sub> N	1051.2	P(16) P(42)	1050.44 1050.49	00 <sup>0</sup> 1-02 <sup>0</sup> 0 00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup> C <sup>12</sup> O <sub>2</sub> <sup>18</sup>		
Ethyl- Pipicolinate	C <sub>8</sub> H <sub>15</sub> NO <sub>2</sub>	1047.3	P(20) P(46)	1046.85 1046.90	00 <sup>0</sup> 1-02 <sup>0</sup> 0 00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup> C <sup>12</sup> O <sub>2</sub> <sup>18</sup>		
Ethyl- Isonipecotate	C <sub>8</sub> H <sub>15</sub> NO <sub>2</sub>	1045.7	P(22) P(48)	1045.02 1045.08	00 <sup>0</sup> 1-02 <sup>0</sup> 0 00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup> C <sup>12</sup> O <sub>2</sub> <sup>18</sup>		
							Soil (Inter- ferent)	
33% Mont- morillonite	Al <sub>1.7</sub> Mg <sub>0.33</sub> [(OH) <sub>2</sub> Si <sub>4</sub> O <sub>10</sub> ]	1040 (± 3)	P(28) P(54) R(32)	1039.37	00 <sup>0</sup> 1-02 <sup>0</sup> 0	C <sup>12</sup> O <sub>2</sub> <sup>16</sup>		
33% Kaolin	Na <sub>0.33</sub> (H <sub>2</sub> O) <sub>4</sub>				1039.50	00 <sup>0</sup> 1-02 <sup>0</sup> 0		C <sup>12</sup> O <sub>2</sub> <sup>18</sup>
34% Illite	Al <sub>4</sub> [(OH) <sub>2</sub> AlSi <sub>3</sub> O <sub>10</sub> (K, H <sub>3</sub> O) Al <sub>2</sub> (OH) <sub>2</sub> AlSi <sub>3</sub> O <sub>10</sub>				1039.38	00 <sup>0</sup> 1-02 <sup>0</sup> 0		C <sup>13</sup> O <sub>2</sub> <sup>16</sup>

## 7. CONCLUSION

A foundation for applying Amplitude and Phase Sensitive Light Scattering (APSLS) technology toward solution of remote detection problems involving chemical and biological contaminants spread across various surfaces was presented. We are currently proceeding with developing an experimental data base of Mueller matrix elements, and will soon compare these data to theoretical predictions as part of a feasibility study. If these laboratory tests prove that contaminants on a surface can be successfully detected at a distance through selective sets of differential (on-then-off molecular resonance by the analyte) Mueller elements, then development of a prototype sensor can begin. With the data base in hand, optimization of hardware and software components in these ellipsometer systems can lead to a simplified prototype system engineered to a specific class of contaminant compounds.

The potential exists for extending this technology toward solution of other identification and classification problems of interest to the Department of Defense, environmental agencies, academia, and private industry. We will undoubtedly realize and incorporate newer and better hardware and software modifications into the present instrument designs for these expanded applications.

Work proceeds on analysis of special Mueller elemental features in beam wavelength and angle orientation that can uniquely represent the analyte (*in situ*), and once presence is established, quantify it. All susceptible Mueller elements will be scrutinized from oblique-to-normal backscattering polar angles, and over the instrument's laser bandwidth spanning 9.1 - 12.2  $\mu\text{m}$ : the mid IR 'fingerprint' region of many important biological and chemical compounds.

Methods of data processing will be improved, including a neural network architecture with pattern recognition algorithm that will likely be integrated into the ellipsometer's analog data acquisition unit. We will continue to study and improve on our methods of analysis of the Mueller field of elements, as we become more familiar with the technology. Future reports will provide updated progress as we complete debugging the software and hardware instrument components, as the digital data acquisition comes on-line, and as quantitative data becomes available. Revisions on the Full Wave theory to include scattering from non isotropic, many-layered, rough surfaces will be reported later, as will absorption and VCD predictions of complex analyte molecules via the quantum chemistry models.

The end goal of this program is to collect and analyze a comprehensive data base on CBW simulants on surfaces, from it identify the crucial polarization, angle, and wavelength parameters that will specify a less complex engineered prototype version of the experimental system, then development and engineering of that sensor.

## LITERATURE CITED

1. Carrieri, A.H., Ditillo, J.T. and Schlein, M.S., *Depolarized Infrared Reflectance From Dry and Wetted Surfaces*, CRDEC-TR-87084, U.S. Army Chemical Research, Development and Engineering Center, Aberdeen Proving Ground, MD, September 1987, UNCLASSIFIED Report.
2. Renau, J., Cheo, P., and Cooper, H., "Depolarization of Linearly Polarized Waves Back-scattered From Rough Metals and Inhomogeneous Dielectrics," *J. Opt. Soc. Am.*, Vol. 57 Num. 4, 1967.
3. Muller, R.H., "Definitions and Conventions in Ellipsometry," *Surf. Sci.*, Vol. 16 Num. 14, 1969.
4. Shurcliff, W.A., *Polarized Light*, Harvard University Press, Cambridge, MA, 1962.
5. Zaininger, K.H., and Revesz, A.G., "Ellipsometry-A Valuable Tool in Surface Research." *RCA Laboratories Review*, pp. 85-115, Princeton, NJ, March 1964.
6. Hauge, P.S., "Conventions and Formulas For Using the Mueller-Stokes Calculus in Ellipsometry," *Surf. Sci.*, Vol. 96 Num. 81, 1980.
7. Simmons, J.W. and Guttman, M.J., *States, Waves and Photons: A Modern Introduction to Light*, Addison-Wesley Publishing Company, Reading, MA, 1970.
8. van de Hulst, H.C., *Light Scattering by Small Particles*, John Wiley and Sons, Inc., New York, NY, 1957.
9. Bohren, C.F. and Huffman, D.R., *Absorption and Scattering of Light by Small Particles*, John Wiley and Sons, Inc., New York, NY, 1983.
10. Jackson, J.D. *Classical Electrodynamics*, Second Edition, Chapter 7, Section 2, John Wiley and Sons, Inc., New York, NY, 1975.
11. Thompson, R.C., Bottiger, J.R., and Fry E.S., "Measurement of Polarized Light Interactions via the Mueller Matrix," *Appl. Opt.*, Vol. 19 Num. 8, 1980.

12. Williams, M.W., "Depolarization and Cross Polarization in Ellipsometry of Rough Surfaces," *Appl. Opt.*, Vol. 25 Num. 20, pp. 3616-22, 1986.

13. Roseler, A., "Use of the Mueller Matrix Formalism for the Spectroscopical IR Ellipsometry with the Fourier Transform Spectrometer," *Optik*, Vol. 61 Num 2, pp. 177-186, 1982.

14. Vorburger, T.V., and Ludema, K.C., "Ellipsometry of Rough Surfaces," *Appl. Opt.*, Vol. 19 Num. 4, pp.561-573, 1980.

15. Fritz Allen and Carlos Bustamante, Editors, *Applications of Circularly Polarized Radiation Using Synchrotron and Ordinary Sources*, Plenum Press, 1985.

Salzman, G.C., et al., "Multiparameter Light Scattering For Rapid Virus Identification," pp. 103-109,

Newman, C.D., and Allen, F.S., "Measuring the Mueller Matrix by a Multimodulator Scattering Instrument," pp. 59-67,

Bickel, W.S., "The Mueller Scattering Matrix Elements For Rayleigh Spheres," pp. 69-76.

16. Kemp, J.C., "Piezo-Optical Birefringence Modulators: New Use for a Long-Known Effect," *JOSA*, Vol. 59 Num. 950, 1969.

17. Hunt, A.J., and Huffman, D.R., "A New Polarization-Modulated Light Scattering Instrument," *Rev. Sci. Instrum.*, Vol. 44, pp. 1753-1762, 1973.

18. Whitt, M.W., and Ulaby, F.T., "Millimeter-Wave Polarimetric Measurements of Artificial and Natural Targets," Conference paper, Inter. Geoscience and Remote Sensing Symposium (IGARSS), Ann Arbor, MI, 18-21 May 1987.

19. van Zyl, Jakob J., and Zebker, H. A., "Interpretation of Imaging Radar Polarization Signatures of Rough Surfaces," Conference Paper, Inter. Geoscience and Remote Sensing Symposium (IGARSS), Ann Arbor, MI, 18-21 May 1987.

20. Mendez, E.R., and O'Donnell, K.A., "Observations of Depolarization and Backscattering Enhancement in Light Scattering From Gaussian Random Surfaces," *Opt. Comm.*, Vol. 61, pp. 91-95, 1987.

21. Query, M., *Optical Constants of Minerals and Other Materials From the Millimeter to the Ultraviolet*, CRDEC-CR-88009, U.S. Army Chemical Research, Development and Engineering Center, Aberdeen Proving Ground, MD, November 1987, UNCLASSIFIED Report.

22. Herzberg, G., *Molecular Spectra and Molecular Structure, Vol. 2, Infrared and Raman Spectra of Polyatomic Molecules*, Van Nostrand, Princeton, NJ, 1945.
23. Tyte, D. C., *Carbon Dioxide Lasers, Advances in Quantum Electronics, Vol. 1*, Goodwin, D. W., Ed., Academic Press, London, 1970.
24. Gradshteyn, I. S., and Ryzhik, I. M., *Table of Integrals, Series, and Products, Equation 8.511(4)*, Academic Press, New York, 1980.
25. Frisch, M., *Gaussian90 User's Guide and Programmer's Reference*, Gaussian Inc., 1990.
26. Amos, R.D., *User's Guide to the Cambridge Analytic Derivative Package*, Cambridge University, 1991.
27. Chabowski, C.F., Jensen, J.O., Yarkony, D.R., and Lengsfeld, B.H., "Theoretical Study of the Radiative Lifetime for the Spin-Forbidden Transition  $a^3\Sigma_u^+ \rightarrow X^1\Sigma_g^+$  in  $\text{He}_2$ ," *J. Chem. Phys.*, Vol. 90 Num. 2504, 1989.
28. Stevens, P.J., "Theory of Vibrational Circular Dichroism," *J. Chem. Phys.* Vol. 89 Num. 661, 1985.
29. Hameka, H.F., "Theoretical Prediction of Structures and Infrared Frequencies," *J. Mol. Struct. (Theochem)* Vol. 226 Num. 241, 1991.
30. Bahar, E., "Review of the Full Wave Solutions for Rough Surface Scattering and Depolarization: Comparisons with Geometric and Physical Optics, Perturbation, and Two-scale Hybrid Solutions," *Journal of Geophysical Research*, Vol. 92 Num. C5, pp. 5209-5224, May 1987.
31. Bahar, E., "Full-Wave Solutions for the Depolarization of the Scattered Radiation Fields by Rough Surfaces of Arbitrary Slope," *IEEE Transactions on Antennas and Propagation*, Vol. AP-29 Num. 3, pp. 443-454, May 1981.
32. Bahar, E. and Fitzwater, M. A., "Scattering Cross Sections for Composite Rough Surfaces Using the Unified Full Wave Approach," *IEEE Transactions on Antennas and Propagation*, Vol. AP-32 Num. 7, pp. 730-734, July 1984.
33. Bahar, E. and Fitzwater, M. A., "Copolarized and Cross-Polarized Incoherent Specific Intensities for Waves at Oblique Incidence Upon a Layer of Finitely Conducting Particles with Rough Surfaces," *Journal of the Optical Society of America A.*, Vol. 4, pp. 41-56, January 1987.

34. Bahar, E., *Scattering and Depolarization of Electromagnetic Waves – Full Wave Solutions*, RADC-TR-83-118, Rome Air Development Center, Air Force Systems Command, Griffiss Air Force Base, NY, November 1983.
35. Morrison, D., *Multivariate Statistical Methods*, Chapters 2,3, and 4, McGraw-Hill, 1967.
36. Carrieri, A.H., "Infrared Detection of Liquids on Terrestrial Surfaces by CO<sub>2</sub> Laser Heating," *Appl. Opt.* Vol. 20 Num. 33, 1990.
37. Rumelhart, McClelland, and the PDP Research Group, *Parallel and Distributed Processing*, MIT Press, pp. 322-330, 1986.
38. Lippmann, R. P., "An Introduction to Computing With Neural Nets," *IEEE ASSP Magazine*, April 1987.

***APPENDIX I. EMISSIONS FROM CO<sub>2</sub> PROBE LASERS***

## APPENDIX I. EMISSIONS FROM CO<sub>2</sub> PROBE LASERS

The following data are reproduced from the **Handbook of Laser Science and Technology**, Marvin J. Weber, CRC Press, Inc. 1982. The laser emissions shown span all allowable wavelength outputs from the ellipsometer's four laser sources. Band nomenclature is of the form  $(\nu_1, \nu_2, \nu_3)$ , where  $\nu_1$  and  $\nu_3$  are quantum numbers specifying stretching modes, and  $\nu_2$  is a quantum number for the bending mode of the linear triatomic CO<sub>2</sub> molecule. The vibrational angular momentum quantum number is specified by l. P(J) and R(J) are vibrational-rotational transitions of the types  $(\nu+1, J-1) \rightarrow (\nu, J)$  and  $(\nu+1, J+1) \rightarrow (\nu, J)$ , respectively, where J is the rotational quantum number. For further details, see the standard text by Herzberg.<sup>20</sup> Each of the ellipsometer's four laser sources can produce between 45 and 75 distinct spectral emissions, most with sufficient power for MCT detection of scattered light from surfaces at ranges to  $\approx 100$  meters at oblique scattering angles and f/10 receiver optics.

## Appendix I

LASER	BAND	TRANSITION	WAVELENGTH	FREQUENCY
			$\mu m$	$cm^{-1}$
$C^{12}O_2^{16}$	$00^0_1 \rightarrow 02^0_0$	R(62)	9.09349	1099.6872
		R(60)	9.09976	1098.9301
		R(58)	9.10623	1098.14940
		R(56)	9.11291	1097.344886
		R(54)	9.11979	1096.516356
		R(52)	9.12689	1095.663612
		R(50)	9.13420	1094.786462
		R(48)	9.14173	1093.884721
		R(46)	9.14948	1092.958211
		R(44)	9.15745	1092.006758
		R(42)	9.16565	1091.030196
		R(40)	9.17407	1090.028367
		R(38)	9.18273	1089.001119
		R(36)	9.19161	1087.948306
		R(34)	9.20073	1086.869791
		R(32)	9.21009	1085.765445
		R(30)	9.21969	1084.635145
		R(28)	9.22953	1083.478778
		R(26)	9.23961	1082.296237
		R(24)	9.24995	1081.087426
		R(22)	9.26053	1079.852255
		R(20)	9.27136	1078.590644
		R(18)	9.28244	1077.302520
		R(16)	9.29379	1075.987820
		R(14)	9.30539	1074.646490
		R(12)	9.31725	1073.278484
		R(10)	9.32937	1071.883766
		R(8)	9.34176	1070.462308
		R(6)	9.35441	1069.024093
		R(4)	9.36734	1067.539110
		R(2)	9.38053	1066.037360
		R(0)	9.39400	1064.508853
		P(2)	9.41472	1062.165965
		P(4)	9.42889	1060.570666
		P(6)	9.44333	1058.948714
		P(8)	9.45805	1057.300161
		P(10)	9.47306	1055.625068
		P(12)	9.48835	1053.923503
		P(14)	9.50394	1052.195545
		P(16)	9.51981	1050.441282

## Appendix I

LASER	BAND	TRANSITION	WAVELENGTH $\mu m$	FREQUENCY $cm^{-1}$
$C^{12}O_2^{16}$	00 <sup>0</sup> 1 - 02 <sup>0</sup> 0	P(18)	9.53597	1048.660810
		P(20)	9.55243	1046.854234
		P(22)	9.56918	1045.021670
		P(24)	9.58623	1043.163239
		P(26)	9.60357	1041.279074
		P(28)	9.62122	1039.369315
		P(30)	9.63917	1037.434110
		P(32)	9.65742	1035.473616
		P(34)	9.67597	1033.487999
		P(36)	9.69483	1031.477430
		P(38)	9.71400	1029.442092
		P(40)	9.73348	1027.382171
		P(42)	9.75326	1025.297865
		P(44)	9.77336	1023.189375
		P(46)	9.79377	1021.056912
		P(48)	9.81450	1018.900693
		P(50)	9.83554	1016.720942
		P(52)	9.85690	1014.517888
		P(54)	9.87858	1012.291767
		P(56)	9.90057	1010.042823
		P(58)	9.92289	1007.771302
		P(60)	9.94552	1005.47746
		P(62)	9.96849	1003.1615
		P(64)	9.99177	1000.8238
		P(66)	10.01538	998.4646
	00 <sup>0</sup> 1 - 10 <sup>0</sup> 0	R(62)	10.02591	997.41550
		R(62)	10.02591	997.41550
		R(60)	10.03347	996.66441
		R(58)	10.04132	995.884686
		R(56)	10.04948	995.076610
		R(54)	10.05793	994.240442
		R(52)	10.06668	993.376427
		R(50)	10.07572	992.484803
		R(48)	10.08506	991.565748
		R(46)	10.09469	990.619630
		R(44)	10.10462	989.646506
		R(42)	10.11484	988.646626
		R(40)	10.12535	987.620181
		R(38)	10.13616	986.567352
		R(36)	10.14725	985.488312

## Appendix I

LASER	BAND	TRANSITION	WAVELENGTH $\mu m$	FREQUENCY $cm^{-1}$
$C^{12}O_2^{16}$	$00^0_1 - 10^0_0$	R(34)	10.15865	984.383226
		R(32)	10.17033	983.252249
		R(30)	10.18231	982.095531
		R(28)	10.19458	980.913211
		R(26)	10.20715	979.705421
		R(24)	10.22001	978.472286
		R(22)	10.23317	977.213922
		R(20)	10.24663	975.930439
		R(18)	10.26039	974.621939
		R(16)	10.27445	973.288517
		R(14)	10.28880	971.930258
		R(12)	10.30347	970.547244
		R(10)	10.31843	969.139547
		R(8)	10.33370	967.707233
		R(6)	10.34928	966.250361
		R(4)	10.36518	964.768982
		R(2)	10.38138	963.263140
		R(0)	10.39790	961.732874
		P(2)	10.42327	959.391745
		P(4)	10.44059	957.800537
		P(6)	10.45823	956.184982
		P(8)	10.47619	954.545087
		P(10)	10.49449	952.880850
		P(12)	10.51312	951.192264
		P(14)	10.53209	949.479314
		P(16)	10.55140	947.741979
		P(18)	10.57105	945.980230
		P(20)	10.59104	944.194030
		P(22)	10.61139	942.383336
		P(24)	10.63210	940.548098
		P(26)	10.65316	938.688257
		P(28)	10.67459	936.803747
		P(30)	10.69639	934.894496
		P(32)	10.71857	932.960421
		P(34)	10.74112	931.001434
		P(36)	10.76406	929.017437
		P(38)	10.78739	927.008325
		P(40)	10.81111	924.973985
		P(42)	10.83524	922.914294
		P(44)	10.85978	920.829123

## Appendix T

LASER	BAND	TRANSITION	WAVELENGTH $\mu m$	FREQUENCY $cm^{-1}$
$C^{12}O_2^{16}$	00 <sup>0</sup> 1 - 10 <sup>0</sup> 0	P(46)	10.88473	918.718331
		P(48)	10.91010	916.581770
		P(50)	10.93590	914.419283
		P(52)	10.96214	912.230703
		P(54)	10.98882	910.015853
		P(56)	11.01595	907.774549
		P(58)	11.04354	905.50659
		P(60)	11.07160	903.21177
		P(62)	11.10014	900.88992
		P(64)	11.12915	898.54082
		P(66)	11.15867	896.1643
		P(68)	11.18868	893.7602
$C^{12}O_2^{18}$	00 <sup>0</sup> 1 - 02 <sup>0</sup> 0	R(50)	8.98767	1112.635004
		R(48)	8.99495	1112.635004
		R(46)	8.99495	1111.735484
		R(42)	9.00238	1110.817288
		R(44)	9.00998	1109.880340
		R(42)	9.01775	1108.924564
		R(40)	9.02568	1107.949890
		R(38)	9.03378	1106.956250
		R(36)	9.04205	1105.943579
		R(34)	9.05050	1104.911817
		R(32)	9.05911	1103.860906
		R(30)	9.06790	1102.790794
		R(28)	9.07687	1101.701429
		R(26)	9.08601	1100.592768
		R(24)	9.09533	1099.464767
		R(22)	9.10484	1098.317390
		R(20)	9.11452	1097.150603
		R(18)	9.12438	1095.964378
		R(16)	9.13443	1094.758688
		R(14)	9.14467	1093.533515
		R(12)	9.15509	1092.288842
		R(10)	9.16570	1091.024658
		R(8)	9.17649	1089.740957
		R(6)	9.18748	1088.437736
		R(4)	9.19866	1087.114998
		R(2)	9.21003	1085.772750
		P(2)	9.23931	1082.331864

## Appendix I

LASER	BAND	TRANSITION	WAVELENGTH	FREQUENCY
			$\mu m$	$cm^{-1}$
$C^{12}O_2^{18}$	00 <sup>0</sup> 1 - 02 <sup>0</sup> 0	P(4)	9.25137	1080.921457
		P(6)	9.26362	1079.491631
		P(8)	9.27607	1078.042418
		P(10)	9.28873	1076.573857
		P(12)	9.30158	1075.085991
		P(14)	9.31464	1073.578866
		P(16)	9.32790	1072.052534
		P(18)	9.34137	1070.507051
		P(20)	9.35504	1068.942477
		P(22)	9.36892	1067.358878
		P(24)	9.38301	1065.756323
		P(26)	9.39730	1064.134886
		P(28)	9.41181	1062.494644
		P(30)	9.42653	1060.835680
		P(32)	9.44146	1059.158080
		P(34)	9.45661	1057.461932
		P(36)	9.47196	1055.747333
		P(38)	9.48754	1054.014378
		P(40)	9.50333	1052.263168
		P(42)	9.51933	1050.493809
		P(44)	9.53556	1048.706409
		P(46)	9.55200	1046.901078
		P(48)	9.56866	1045.077932
		P(50)	9.58555	1043.237087
		P(52)	9.60265	1041.378663
		P(54)	9.61998	1039.502785
		P(56)	9.63754	1037.609577
		P(58)	9.65531	1035.699167
	00 <sup>0</sup> 1 - 10 <sup>0</sup> 0	R(44)	10.08328	991.274098
		R(42)	10.09604	990.487703
		R(40)	10.10435	989.673232
		R(38)	10.11295	988.830811
		R(36)	10.12186	987.960562
		R(34)	10.13107	987.062600
		R(32)	10.14058	986.137035
		R(30)	10.15039	985.183973
		R(28)	10.16050	984.203513
		R(26)	10.17091	983.195749
		R(24)	10.18163	982.160770
		R(22)	10.19265	981.098661

## Appendix I

LASER	BAND	TRANSITION	WAVELENGTH $\mu m$	FREQUENCY $cm^{-1}$
$C^{12}O_2^{18}$	$00^0_1 - 10^0_0$	R(20)	10.20398	980.009499
		R(18)	10.21562	978.893358
		R(16)	10.22756	977.750307
		R(14)	10.23981	976.580410
		R(12)	10.25238	975.383724
		R(10)	10.26525	974.160302
		R(8)	10.27844	972.910195
		R(6)	10.29195	971.633444
		R(4)	10.30577	970.330089
		P(6)	10.38759	962.687339
		P(8)	10.40354	961.211656
		P(10)	10.41982	959.709502
		P(12)	10.43644	958.180873
		P(14)	10.45341	956.625761
		P(16)	10.47072	955.044153
		P(18)	10.48838	953.436031
		P(20)	10.50639	951.801372
		P(22)	10.52476	950.140149
		P(24)	10.54349	948.452326
		P(26)	10.56259	946.737867
		P(28)	10.58205	944.996728
		P(30)	10.60188	943.228860
		P(32)	10.62209	941.434209
		P(34)	10.64268	939.612716
		P(36)	10.66366	937.764316
		P(38)	10.68503	935.888939
		P(40)	10.70679	933.986510
		P(42)	10.72896	932.056949
		P(44)	10.75153	930.100167
		P(46)	10.77451	928.116074
		P(48)	10.79792	926.104570
$C^{13}O_2^{16}$	$00^0_1 - 02^0_0$	R(36)	9.60169	1041.483334
		R(34)	9.61126	1040.446675
		R(32)	9.62110	1039.381840
		R(30)	9.63123	1038.288700
		R(28)	9.64165	1037.167135
		R(26)	9.65235	1036.017032
		R(24)	9.66335	1034.838287

## Appendix T

LASER	BAND	TRANSITION	WAVELENGTH	FREQUENCY
			$\mu\text{m}$	$\text{cm}^{-1}$
$\text{C}^{13}\text{O}_2^{16}$	$00^01 - 02^00$	R(22)	9.67463	1033.630806
		R(20)	9.68622	1032.394502
		R(18)	9.69810	1031.129298
		R(16)	9.71029	1029.835128
		R(14)	9.72278	1028.511931
		R(12)	9.73558	1027.159658
		R(10)	9.74870	1025.778270
		R(8)	9.76212	1024.367737
		R(6)	9.77586	1022.928037
		R(4)	9.78992	1021.459160
		P(6)	9.87304	1012.859224
		P(8)	9.88923	1011.201098
		P(10)	9.90576	1009.514024
		P(12)	9.92262	1007.798072
		P(14)	9.93983	1006.053323
		P(16)	9.95738	1004.279869
		P(18)	9.97528	1002.477810
		P(20)	9.99353	1000.647256
		P(22)	10.01213	998.788325
		P(24)	10.03108	996.901145
		P(26)	10.05039	994.985854
		P(28)	10.07006	993.042598
		P(30)	10.09009	991.071531
		P(32)	10.11048	989.072816
		P(34)	10.13123	987.046625
		P(36)	10.15235	984.993138
		P(38)	10.17385	982.912542
	$00^01 - 10^00$	R(44)	10.60063	943.340303
		R(42)	10.61310	942.231411
		R(40)	10.62585	941.101238
		R(38)	10.63886	939.949924
		R(36)	10.65215	938.777604
		R(34)	10.66571	937.584403
		R(32)	10.67953	936.370443
		R(30)	10.69363	935.135838
		R(28)	10.70801	933.880687
		R(26)	10.72265	932.605121
		R(24)	10.73757	931.309207
		R(22)	10.75277	929.993046

## Appendix I

LASER	BAND	TRANSITION	WAVELENGTH $\mu m$	FREQUENCY $cm^{-1}$
$C^{13}O_2^{16}$	$00^0_1 - 10^0_0$	R(20)	10.76824	928.656723
		R(18)	10.78399	927.300318
		R(16)	10.80002	925.923906
		R(14)	10.81634	924.527554
		R(12)	10.83293	923.111328
		R(10)	10.84981	921.675286
		R(8)	10.86697	920.219482
		R(6)	10.88443	918.743964
		R(4)	10.90217	917.248777
		P(4)	10.98566	910.277955
		P(6)	11.00503	908.675151
		P(8)	11.02472	907.052844
		P(10)	11.04471	905.411040
		P(12)	11.06501	903.749742
		P(14)	11.08563	902.068947
		P(16)	11.10656	900.368647
		P(18)	11.12782	898.648830
		P(20)	11.14940	896.909477
		P(22)	11.17131	895.150565
		P(24)	11.19534	893.372066
		P(26)	11.21612	891.573944
		P(28)	11.23903	889.756160
		P(30)	11.26229	887.918669
		P(32)	11.28590	886.061419
		P(34)	11.30986	884.184353
		P(36)	11.33418	882.287407
		P(38)	11.35885	880.370512
		P(40)	11.38390	878.433591
		P(42)	11.40932	876.476562
		P(44)	11.43511	874.49933
		P(46)	11.46129	872.50181
		P(48)	11.48786	870.48389
$C^{14}O_2^{16}$	$00^0_1 - 02^0_0$	R(40)	9.91788	1008.280282
		R(38)	9.92733	
		R(36)	9.93709	1006.330912
		R(34)	9.94715	1005.312772
		R(32)	9.95753	1004.265463
		R(30)	9.96821	1003.188845

Appendix I

LASER	BAND	TRANSITION	WAVELENGTH $\mu m$	FREQUENCY $cm^{-1}$
$C^{14}O_2^{16}$	$00^0_1 \rightarrow 02^0_0$	R(28)	9.97922	1002.082785
		R(26)	9.99054	1000.947161
		R(24)	10.00218	999.781858
		R(22)	10.01415	998.586771
		R(20)	10.02645	997.361804
		R(18)	10.03908	996.106870
		R(16)	10.05205	994.821893
		R(14)	10.06536	993.506806
		R(12)	10.07900	992.161553
		R(10)	10.09300	990.786087
		R(8)	10.10734	998.380373
		R(6)	10.12203	987.944385
		P(8)	10.24370	976.209763
		P(10)	10.26149	974.516934
		P(12)	10.27967	972.794124
		P(14)	10.29822	971.041421
		P(16)	10.31716	969.258921
		P(18)	10.33649	967.446731
		P(20)	10.35620	965.604971
		P(22)	10.37631	963.733766
		P(24)	10.39681	961.833254
		P(26)	10.41771	959.903583
		P(28)	10.43901	957.944909
		P(30)	10.46072	955.957396
		P(32)	10.48283	953.941220
		P(34)	10.50534	951.896562
		P(36)	10.52827	949.823614
	$00^0_1 \rightarrow 10^0_0$	R(50)	11.10699	900.33358
		R(48)	11.12097	899.20226
		R(46)	11.13520	898.05318
		R(44)	11.14968	896.88643
		R(42)	11.16443	895.70211
		R(40)	11.17943	894.50031
		R(38)	11.19468	893.28113
		R(36)	11.21020	892.04463
		R(34)	11.22598	890.79092
		R(32)	11.24202	889.52005
		R(30)	11.25832	888.23212
		R(28)	11.27488	886.92718

## Appendix I

LASER	BAND	TRANSITION	WAVELENGTH $\mu m$	FREQUENCY $cm^{-1}$
$C^{14}O_2^{16}$	00 <sup>0</sup> 1 - 10 <sup>0</sup> 0	R(26)	11.29171	885.60530
		R(24)	11.30881	884.26654
		R(22)	11.32617	882.91098
		R(18)	11.36170	880.14964
		R(16)	11.37988	878.74397
		R(14)	11.39833	877.32170
		R(12)	11.41705	875.88288
		R(10)	11.43605	874.42754
		R(8)	11.45533	872.95574
		R(6)	11.47490	871.46751
		R(4)	11.49474	869.96288
		P(6)	11.60907	861.39566
		P(8)	11.63081	859.78513
		P(10)	11.65286	858.15839
		P(12)	11.67521	856.51545
		P(14)	11.69787	854.85631
		P(16)	11.72084	853.18100
		P(18)	11.74413	851.48950
		P(20)	11.76773	849.78182
		P(22)	11.79165	848.05797
		P(24)	11.81589	846.31794
		P(26)	11.84046	844.56172
		P(28)	11.86536	842.78930
		P(30)	11.89060	841.00067
		P(32)	11.91617	839.19581
		P(34)	11.94209	837.37471
		P(36)	11.96835	835.53734
		P(38)	11.99496	833.68367
		P(40)	12.02192	831.81368
		P(42)	12.04925	829.92733
		P(44)	12.07694	828.02458
		P(46)	12.10499	826.10540
		P(48)	12.13342	824.16974

**APPENDIX II. SYSTEM TO SAMPLE MUELLER MATRIX TRANSFORMATION: 3-MIRROR  
GONIOMETER TYPE ELLIPSOMETER FOR IN-SITU ANALYSES OF POROUS SURFACES.**

## APPENDIX II. SYSTEM TO SAMPLE MUELLER MATRIX TRANSFORMATION: 3-MIRROR GONIOMETER TYPE ELLIPSOMETER FOR IN-SITU ANALYSES OF POROUS SURFACES.

The sixteen equations that follow decouple Mueller matrix elements of the scattering sample ( $f_{ij}$ ) from *system* matrix elements (Equation 6b,  $\psi_{ij}$ ) generated by the ellipsometer facility of Figure 4a. The total scattering signal includes contributions from the sample and four flat mirrors oriented at  $45^\circ$  incidence, and positioned between transmission- and collection-beam PEM modulators.

Elements  $b_{ij}$ ,  $c_{ij}$ , and  $d_{ij}$  are experimental calibration data as measured from the configurations schematically drawn in Figure 10, and  $\theta$  is the angle of backscattering subtended by the goniometer arm. MACSYMA, an interactive symbolic mathematical program, was used in determining the product of the ten  $4 \times 4$  matrices of Equation 6b, and in producing the Fortran code of the following equations for use in the ellipsometer's data acquisition system.

It suffices, from the experimental complexity and uncertainty of this calibration procedure, that an optical redesign that self-compensates all mirror elements is more practical as a means of accomplishing sample Mueller element measurements from terrestrial surfaces. (See Section 4.6.4.)

# Appendix II

$$\begin{pmatrix} \psi_{11} & \psi_{12} & \psi_{13} & \psi_{14} \\ \psi_{21} & \psi_{22} & \psi_{23} & \psi_{24} \\ \psi_{31} & \psi_{32} & \psi_{33} & \psi_{34} \\ \psi_{41} & \psi_{42} & \psi_{43} & \psi_{44} \end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ f_{41} & f_{42} & f_{43} & f_{44} \end{pmatrix}$$

$$f_{44}(\theta) = \frac{\psi_{44}}{b_{33}d_{33}}$$

$$f_{43}(\theta) = -\frac{\psi_{42}\sin(2\theta) - \psi_{43}\cos(2\theta)}{b_{33}d_{33}}$$

$$f_{42}(\theta) = -\frac{b_{11}\psi_{43}\sin(2\theta) + b_{11}\psi_{42}\cos(2\theta) - b_{12}\psi_{41}}{(b_{12}^2 - b_{11}^2)d_{33}}$$

$$f_{41}(\theta) = \frac{b_{12}\psi_{43}\sin(2\theta) + b_{12}\psi_{42}\cos(2\theta) - b_{11}\psi_{41}}{(b_{12}^2 - b_{11}^2)d_{33}}$$

$$f_{34}(\theta) = [((c_{12}d_{12} - c_{11}d_{11})d_{33}\psi_{24} + (c_{11}d_{12} - c_{12}d_{11})d_{33}\psi_{14})\sin(2\theta) + (c_{33}d_{12}^2 -$$

$$\frac{c_{33}d_{11}^2)\psi_{34}\cos(2\theta)]}{(b_{33}c_{33}d_{12}^2 - b_{33}c_{33}d_{11}^2)d_{33}}$$

$$f_{33}(\theta) = -[((c_{12}d_{12} - c_{11}d_{11})d_{33}\psi_{22} + (c_{11}d_{12} - c_{12}d_{11})d_{33}\psi_{12})\sin^2(2\theta) + ((c_{33}d_{12}^2 -$$

$$\frac{(c_{33}d_{11}^2 - c_{33}d_{12}^2)\psi_{33}\cos^2(2\theta)]}{(b_{33}c_{33}d_{12}^2 - b_{33}c_{33}d_{11}^2)d_{33}}$$

Appendix II

$$f_{32}(\theta) = -[((b_{11}c_{12}d_{12}-b_{11}c_{11}d_{11})d_{33}\psi_{23}+(b_{11}c_{11}d_{12}-b_{11}c_{12}d_{11})d_{33}\psi_{13})\sin^2(2\theta)+$$

$$(((b_{11}c_{33}d_{12}^2-b_{11}c_{33}d_{11}^2)\psi_{33}+(b_{11}c_{12}d_{12}-b_{11}c_{11}d_{11})d_{33}\psi_{22}+(b_{11}c_{11}d_{12}-$$

$$b_{11}c_{12}d_{11})d_{33}\psi_{12})\cos(2\theta)+(b_{12}c_{11}d_{11}-b_{12}c_{12}d_{12})d_{33}\psi_{21}+(b_{12}c_{12}d_{11}-$$

$$b_{12}c_{11}d_{12})d_{33}\psi_{11})\sin(2\theta)+(b_{11}c_{33}d_{12}^2-b_{11}c_{33}d_{11}^2)\psi_{32}\cos^2(2\theta)+$$

$$\frac{(b_{12}c_{33}d_{11}^2-b_{12}c_{33}d_{12}^2)\psi_{31}\cos(2\theta)]}{((b_{12}^2-b_{11}^2)c_{33}d_{12}^2+(b_{11}^2-b_{12}^2)c_{33}d_{11}^2)d_{33}}$$

$$f_{31}(\theta) = [((b_{12}c_{12}d_{12}-b_{12}c_{11}d_{11})d_{33}\psi_{23}+(b_{12}c_{11}d_{12}-b_{12}c_{12}d_{11})d_{33}\psi_{13})\sin^2(2\theta)+$$

$$(((b_{12}c_{33}d_{12}^2-b_{12}c_{33}d_{11}^2)\psi_{33}+(b_{12}c_{12}d_{12}-b_{12}c_{11}d_{11})d_{33}\psi_{22}+(b_{12}c_{11}d_{12}-$$

$$b_{12}c_{12}d_{11})d_{33}\psi_{12})\cos(2\theta)+(b_{11}c_{11}d_{11}-b_{11}c_{12}d_{12})d_{33}\psi_{21}+(b_{11}c_{12}d_{11}-$$

$$b_{11}c_{11}d_{12})d_{33}\psi_{11})\sin(2\theta)+(b_{12}c_{33}d_{12}^2-b_{12}c_{33}d_{11}^2)\psi_{32}\cos^2(2\theta)+(b_{11}c_{33}d_{11}^2-$$

$$\frac{b_{11}c_{33}d_{12}^2)\psi_{31}\cos(2\theta)]}{((b_{12}^2-b_{11}^2)c_{33}d_{12}^2+(b_{11}^2-b_{12}^2)c_{33}d_{11}^2)d_{33}}$$

$$f_{24}(\theta) = [(c_{11}c_{33}d_{12}^2-c_{11}c_{33}d_{11}^2)\psi_{34}\sin(2\theta)+((c_{11}^2d_{11}-c_{11}c_{12}d_{12})d_{33}\psi_{24}+$$

$$(c_{11}c_{12}d_{11}-c_{11}^2d_{12})d_{33}\psi_{14})\cos(2\theta)+(c_{11}c_{12}d_{12}-c_{11}^2d_{11})d_{33}\psi_{24}+$$

$$\frac{(c_{12}^2d_{12}-c_{11}c_{12}d_{11})d_{33}\psi_{14}]}{(b_{33}c_{12}^2-b_{33}c_{11}^2)d_{12}^2+(b_{33}c_{11}^2-b_{33}c_{12}^2)d_{11}^2)d_{33}}$$

$$f_{23}(\theta) = -[(c_{11}c_{33}d_{12}^2-c_{11}c_{33}d_{11}^2)\psi_{32}\sin^2(2\theta)+((c_{11}c_{33}d_{11}^2-$$

$$c_{11}c_{33}d_{12}^2)\psi_{33}+(c_{11}^2d_{11}-c_{11}c_{12}d_{12})d_{33}\psi_{22}+(c_{11}c_{12}d_{11}-$$

$$c_{11}^2d_{12})d_{33}\psi_{12})\cos(2\theta)+(c_{11}c_{12}d_{12}-c_{11}^2d_{11})d_{33}\psi_{22}+(c_{12}^2d_{12}-$$

$$c_{11}c_{12}d_{11})d_{33}\psi_{12})\sin(2\theta)+((c_{11}c_{12}d_{12}-c_{11}^2d_{11})d_{33}\psi_{23}+(c_{11}^2d_{12}-$$

$$c_{11}c_{12}d_{11})d_{33}\psi_{13})\cos^2(2\theta)+((c_{12}^2d_{11}-c_{11}c_{12}d_{12})d_{33}\psi_{23}+(c_{11}c_{12}d_{11}-$$

$$\frac{c_{12}^2d_{12})d_{33}\psi_{13})\cos(2\theta)]}{((b_{33}c_{12}^2-b_{33}c_{11}^2)d_{12}^2+(b_{33}c_{11}^2-b_{33}c_{12}^2)d_{11}^2)d_{33}}$$

$$\begin{aligned}
 f_{22}(\theta) = & -[(b_{11}c_{11}c_{33}d_{12}^2 - b_{11}c_{11}c_{33}d_{11}^2)\psi_{33}\sin^2(2\theta) + ((b_{11}c_{11}c_{33}d_{12}^2 - \\
 & b_{11}c_{11}c_{33}d_{11}^2)\psi_{32} + (b_{11}c_{11}^2d_{11} - b_{11}c_{11}c_{12}d_{12})d_{33}\psi_{23} + (b_{11}c_{11}c_{12}d_{11} - \\
 & b_{11}c_{11}^2d_{12})d_{33}\psi_{13})\cos(2\theta) + (b_{12}c_{11}c_{33}d_{11}^2 - b_{12}c_{11}c_{33}d_{12}^2)\psi_{31} + \\
 & (b_{11}c_{11}c_{12}d_{12} - b_{11}c_{12}^2d_{11})d_{33}\psi_{23} + (b_{11}c_{12}^2d_{12} - b_{11}c_{11}c_{12}d_{11})d_{33}\psi_{13})\sin(2\theta) + \\
 & ((b_{11}c_{11}^2d_{11} - b_{11}c_{11}c_{12}d_{12})d_{33}\psi_{22} + (b_{11}c_{11}c_{12}d_{11} - b_{11}c_{11}^2d_{12})d_{33}\psi_{12})\cos^2(2\theta) + \\
 & ((b_{11}c_{11}c_{12}d_{12} - b_{11}c_{12}^2d_{11})d_{33}\psi_{22} + (b_{12}c_{11}c_{12}d_{12} - b_{12}c_{11}^2d_{11})d_{33}\psi_{21} + \\
 & (b_{11}c_{12}^2d_{12} - b_{11}c_{11}c_{12}d_{11})d_{33}\psi_{12} + (b_{12}c_{11}^2d_{12} - b_{12}c_{11}c_{12}d_{11})d_{33}\psi_{11})\cos(2\theta) + \\
 & \frac{(b_{12}c_{12}^2d_{11} - b_{12}c_{11}c_{12}d_{12})d_{33}\psi_{21} + (b_{12}c_{11}c_{12}d_{11} - b_{12}c_{12}^2d_{12})d_{33}\psi_{11}}{((b_{12}^2 - b_{11}^2)c_{12}^2 + (b_{11}^2 - b_{12}^2)c_{11}^2)d_{12}^2 + ((b_{11}^2 - b_{12}^2)c_{12}^2 + (b_{12}^2 - b_{11}^2)c_{11}^2)d_{11}^2}d_{33}
 \end{aligned}$$

$$\begin{aligned}
 f_{21}(\theta) = & [(b_{12}c_{11}c_{33}d_{12}^2 - b_{12}c_{11}c_{33}d_{11}^2)\psi_{33}\sin^2(2\theta) + ((b_{12}c_{11}c_{33}d_{12}^2 - \\
 & b_{12}c_{11}c_{33}d_{11}^2)\psi_{32} + (b_{12}c_{11}^2d_{11} - b_{12}c_{11}c_{12}d_{12})d_{33}\psi_{23} + (b_{12}c_{11}c_{12}d_{11} - \\
 & b_{12}c_{11}^2d_{12})d_{33}\psi_{13})\cos(2\theta) + (b_{11}c_{11}c_{33}d_{11}^2 - b_{11}c_{11}c_{33}d_{12}^2)\psi_{31} + \\
 & (b_{12}c_{11}c_{12}d_{12} - b_{12}c_{12}^2d_{11})d_{33}\psi_{23} + (b_{12}c_{12}^2d_{12} - b_{12}c_{11}c_{12}d_{11})d_{33}\psi_{13})\sin(2\theta) + \\
 & ((b_{12}c_{11}^2d_{11} - b_{12}c_{11}c_{12}d_{12})d_{33}\psi_{22} + (b_{12}c_{11}c_{12}d_{11} - \\
 & b_{12}c_{11}^2d_{12})d_{33}\psi_{12})\cos^2(2\theta) + ((b_{12}c_{11}c_{12}d_{12} - b_{12}c_{12}^2d_{11})d_{33}\psi_{22} + (b_{11}c_{11}c_{12}d_{12} - \\
 & b_{11}c_{11}^2d_{11})d_{33}\psi_{21} + (b_{12}c_{12}^2d_{12} - b_{12}c_{11}c_{12}d_{11})d_{33}\psi_{12} + (b_{11}c_{11}^2d_{12} - \\
 & b_{11}c_{11}c_{12}d_{11})d_{33}\psi_{11})\cos(2\theta) + (b_{11}c_{12}^2d_{11} - b_{11}c_{11}c_{12}d_{12})d_{33}\psi_{21} + \\
 & \frac{(b_{11}c_{11}c_{12}d_{11} - b_{11}c_{12}^2d_{12})d_{33}\psi_{11}}{((b_{12}^2 - b_{11}^2)c_{12}^2 + (b_{11}^2 - b_{12}^2)c_{11}^2)d_{12}^2 + ((b_{11}^2 - b_{12}^2)c_{12}^2 + (b_{12}^2 - b_{11}^2)c_{11}^2)d_{11}^2}d_{33}
 \end{aligned}$$

$$\begin{aligned}
 f_{14}(\theta) = & -[(c_{12}c_{33}d_{12}^2 - c_{12}c_{33}d_{11}^2)\psi_{34}\sin(2\theta) + ((c_{11}c_{12}d_{11} - c_{12}^2d_{12})d_{33}\psi_{24} + \\
 & (c_{12}^2d_{11} - c_{11}c_{12}d_{12})d_{33}\psi_{14})\cos(2\theta) + (c_{11}^2d_{12} - c_{11}c_{12}d_{11})d_{33}\psi_{24} + \\
 & \frac{(c_{11}c_{12}d_{12} - c_{11}^2d_{11})d_{33}\psi_{14}}{(b_{33}c_{12}^2 - b_{33}c_{11}^2)d_{12}^2 + (b_{33}c_{11}^2 - b_{33}c_{12}^2)d_{11}^2}d_{33}
 \end{aligned}$$

Appendix II

$$f_{13}(\theta) = [(c_{12}c_{33}d_{12}^2 - c_{12}c_{33}d_{11}^2)\psi_{32}\sin^2(2\theta) + ((c_{12}c_{33}d_{11}^2 - c_{12}c_{33}d_{12}^2)\psi_{33} + (c_{11}c_{12}d_{11} - c_{12}^2d_{12})d_{33}\psi_{22} + (c_{12}^2d_{11} - c_{11}c_{12}d_{12})d_{33}\psi_{12})\cos(2\theta) + (c_{11}^2d_{12} - c_{11}c_{12}d_{11})d_{33}\psi_{22} + (c_{11}c_{12}d_{12} - c_{11}^2d_{11})d_{33}\psi_{12})\sin(2\theta) + ((c_{12}^2d_{12} - c_{11}c_{12}d_{11})d_{33}\psi_{23} + (c_{11}c_{12}d_{12} - c_{12}^2d_{11})d_{33}\psi_{13})\cos^2(2\theta) + \frac{((c_{11}c_{12}d_{11} - c_{11}^2d_{12})d_{33}\psi_{23} + (c_{11}^2d_{11} - c_{11}c_{12}d_{12})d_{33}\psi_{13})\cos(2\theta)}{(b_{33}c_{12}^2 - b_{33}c_{11}^2)d_{12}^2 + (b_{33}c_{11}^2 - b_{33}c_{12}^2)d_{11}^2}d_{33}$$

$$f_{12}(\theta) = [(b_{11}c_{12}c_{33}d_{12}^2 - b_{11}c_{12}c_{33}d_{11}^2)\psi_{33}\sin^2(2\theta) + ((b_{11}c_{12}c_{33}d_{12}^2 - b_{11}c_{12}c_{33}d_{11}^2)\psi_{32} + (b_{11}c_{11}c_{12}d_{11} - b_{11}c_{12}^2d_{12})d_{33}\psi_{23} + (b_{11}c_{12}^2d_{11} - b_{11}c_{11}c_{12}d_{12})d_{33}\psi_{13})\cos(2\theta) + (b_{12}c_{12}c_{33}d_{11}^2 - b_{12}c_{12}c_{33}d_{12}^2)\psi_{31} + (b_{11}c_{11}^2d_{12} - b_{11}c_{11}c_{12}d_{11})d_{33}\psi_{23} + (b_{11}c_{11}c_{12}d_{12} - b_{11}c_{12}^2d_{11})d_{33}\psi_{13})\sin(2\theta) + ((b_{11}c_{11}c_{12}d_{11} - b_{11}c_{12}^2d_{12})d_{33}\psi_{22} + (b_{11}c_{12}^2d_{11} - b_{11}c_{11}c_{12}d_{12})d_{33}\psi_{12})\cos^2(2\theta) + ((b_{11}c_{11}^2d_{12} - b_{11}c_{11}c_{12}d_{11})d_{33}\psi_{22} + (b_{12}c_{12}^2d_{12} - b_{12}c_{11}c_{12}d_{11})d_{33}\psi_{21} + (b_{11}c_{11}c_{12}d_{12} - b_{11}c_{11}^2d_{11})d_{33}\psi_{12} + (b_{12}c_{11}c_{12}d_{12} - b_{12}c_{12}^2d_{11})d_{33}\psi_{11})\cos(2\theta) + \frac{(b_{12}c_{11}c_{12}d_{11} - b_{12}c_{11}^2d_{12})d_{33}\psi_{21} + (b_{12}c_{11}^2d_{11} - b_{12}c_{11}c_{12}d_{12})d_{33}\psi_{11}}{((b_{12}^2 - b_{11}^2)c_{12}^2 + (b_{11}^2 - b_{12}^2)c_{11}^2)d_{12}^2 + ((b_{11}^2 - b_{12}^2)c_{12}^2 + (b_{12}^2 - b_{11}^2)c_{11}^2)d_{11}^2}d_{33}$$

# Appendix II

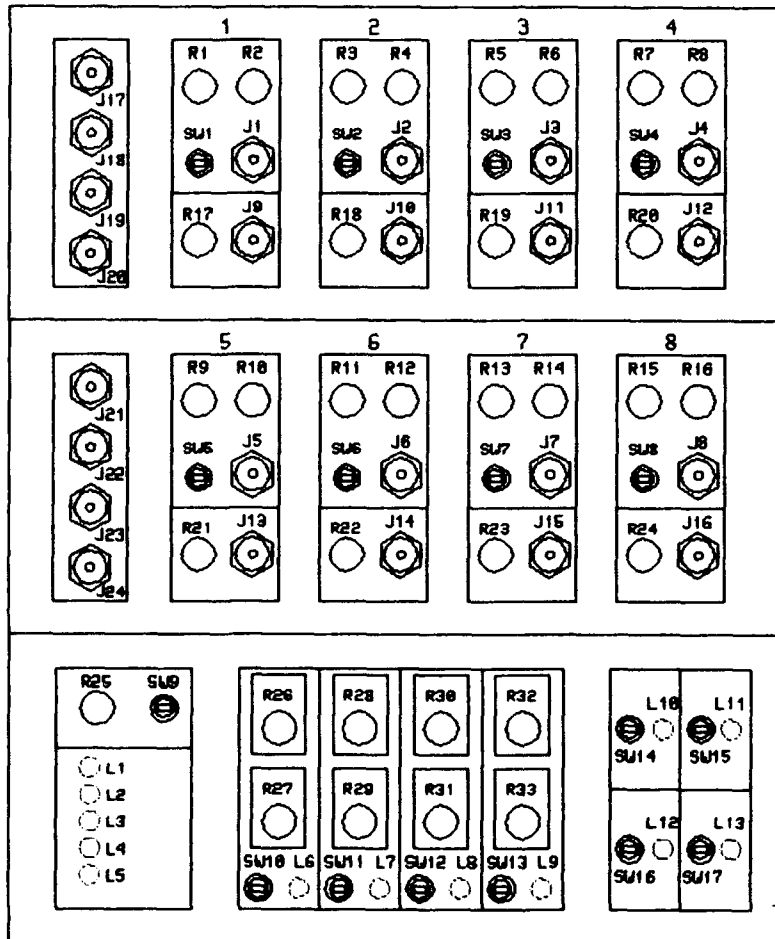
$$\begin{aligned}
 f_{11}(\theta) = & -[(b_{12}c_{12}c_{33}d_{12}^2 - b_{12}c_{12}c_{33}d_{11}^2)\psi_{33}\sin^2(2\theta) + ((b_{12}c_{12}c_{33}d_{12}^2 - \\
 & b_{12}c_{12}c_{33}d_{11}^2)\psi_{32} + (b_{12}c_{11}c_{12}d_{11} - b_{12}c_{12}^2d_{12})d_{33}\psi_{23} + (b_{12}c_{12}^2d_{11} - \\
 & b_{12}c_{11}c_{12}d_{12})d_{33}\psi_{13})\cos(2\theta) + (b_{11}c_{12}c_{33}d_{11}^2 - b_{11}c_{12}c_{33}d_{12}^2)\psi_{31} + \\
 & (b_{12}c_{11}^2d_{12} - b_{12}c_{11}c_{12}d_{11})d_{33}\psi_{23} + (b_{12}c_{11}c_{12}d_{12} - \\
 & b_{12}c_{11}^2d_{11})d_{33}\psi_{13})\sin(2\theta) + ((b_{12}c_{11}c_{12}d_{11} - b_{12}c_{12}^2d_{12})d_{33}\psi_{22} + (b_{12}c_{12}^2d_{11} - \\
 & b_{12}c_{11}c_{12}d_{12})d_{33}\psi_{12})\cos^2(2\theta) + ((b_{12}c_{11}^2d_{12} - b_{12}c_{11}c_{12}d_{11})d_{33}\psi_{22} + (b_{11}c_{12}^2d_{12} - \\
 & b_{11}c_{11}c_{12}d_{11})d_{33}\psi_{21} + (b_{12}c_{11}c_{12}d_{12} - b_{12}c_{11}^2d_{11})d_{33}\psi_{12} + (b_{11}c_{11}c_{12}d_{12} - \\
 & b_{11}c_{12}^2d_{11})d_{33}\psi_{11})\cos(2\theta) + (b_{11}c_{11}c_{12}d_{11} - b_{11}c_{11}^2d_{12})d_{33}\psi_{21} + (b_{11}c_{11}^2d_{11} - \\
 & b_{11}c_{11}c_{12}d_{12})d_{33}\psi_{11}] \\
 & \frac{b_{11}c_{11}c_{12}d_{12}d_{33}\psi_{11}}{((b_{12}^2 - b_{11}^2)c_{12}^2 + (b_{11}^2 - b_{12}^2)c_{11}^2)d_{12}^2 + ((b_{11}^2 - b_{12}^2)c_{12}^2 + (b_{12}^2 - b_{11}^2)c_{11}^2)d_{11}^2)d_{33}}
 \end{aligned}$$

Blank

### *APPENDIX III: APSD ELECTRONIC CIRCUITS. MODULES I-VIII*

The Analog Phase-Sensitive Detector (APSD), designed and fabricated by Dave Owens, is an integrated 8-module electronic system that produces all Mueller elements from the scattergram in a highly automated manner, under the control of a host microVax computer. An experimenter would typically initialize the ellipsometer (select sample, analyte, beam wavelengths, incident backscattering angles) and monitor the automated progress of the APSD by LED readouts on its front control panel. The APSD recognizes the optical configurations (Tables 3-4), acquires the scattergram (Equations 12, 14b, 16b, and 18b), and makes the Mueller element mappings that are digitized, graphed, and stored on disk. We provide in this appendix the major APSD circuits now operating in this ellipsometer system.

# FRONT PANEL LAYOUT

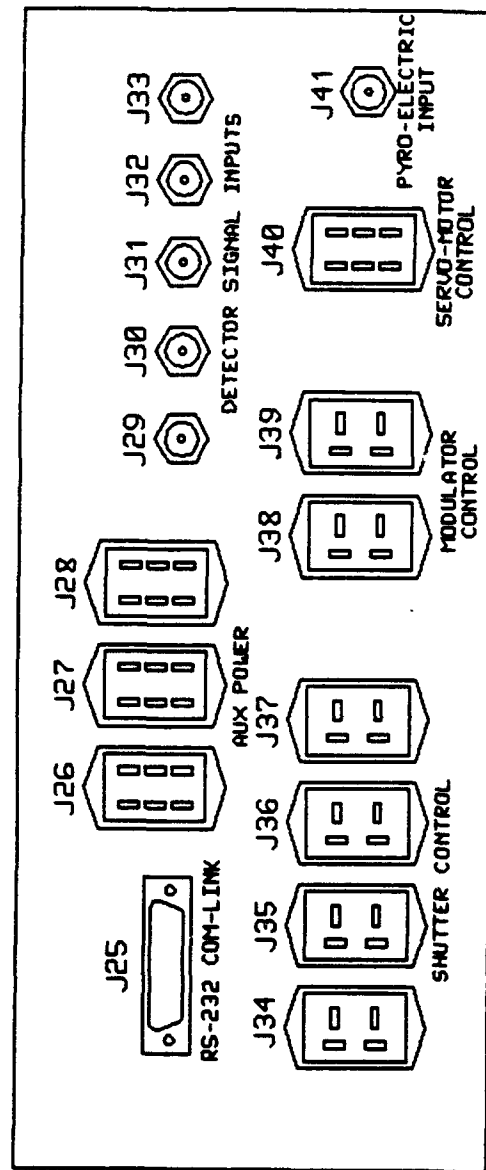


## CORRESPONDING FREQUENCIES

1:	31.896 khz
2:	33.96 khz
3:	63.79 khz
4:	67.92 khz
5:	4.13 khz
6:	65.86 khz
7:	97.75 khz
8:	99.82 khz

Figure AIII.1. The front panel layout of the Analog Phase-Sensitive Detector (APSD). Eight PSD channels are represented (1-8). Connectors J17-J20 and J21-J24 are inputs/outputs of the primary modulator frequencies  $\omega_1$ ,  $\omega_2$ ,  $2\omega_1$ , and  $2\omega_2$ . R1-R16 are the course (odd) and fine (even) reference phase shift controllers, while SW1-SW8 are the phase inverters, per channel. Connectors J1-J8 are reference frequency outputs, while J9-J12 and J13-J16 are the PSD (Mueller elements) output channels. R17-R24 are the PSD amplitude adjustments, while R26-R33 are retardation adjusters to transmitter and receiver PEM's and switches SW10-SW13 allow manual or remote retardation control. LED's L1-L5 are status indicators for the incident beams power regulation circuit, while L10, SW14 - L13, SW17 show operational status of the shutter controllers and allow for manual or remote switching between beams.

## BACK PANEL LAYOUT



J-29, 30      DARK SIGNAL  
 J-31          AGC      DC DETECTOR SIGNAL  
 J-32          DC DETECTOR SIGNAL  
 J-33          AC DETECTOR SIGNAL

Figure AIII.2. The APSD back panel layout. J25 is the serial communications link to and from the host CPU and stepper motor controllers, A/D converter, modulator and servo-motor controls, shutter and pyro-electric detector devices.

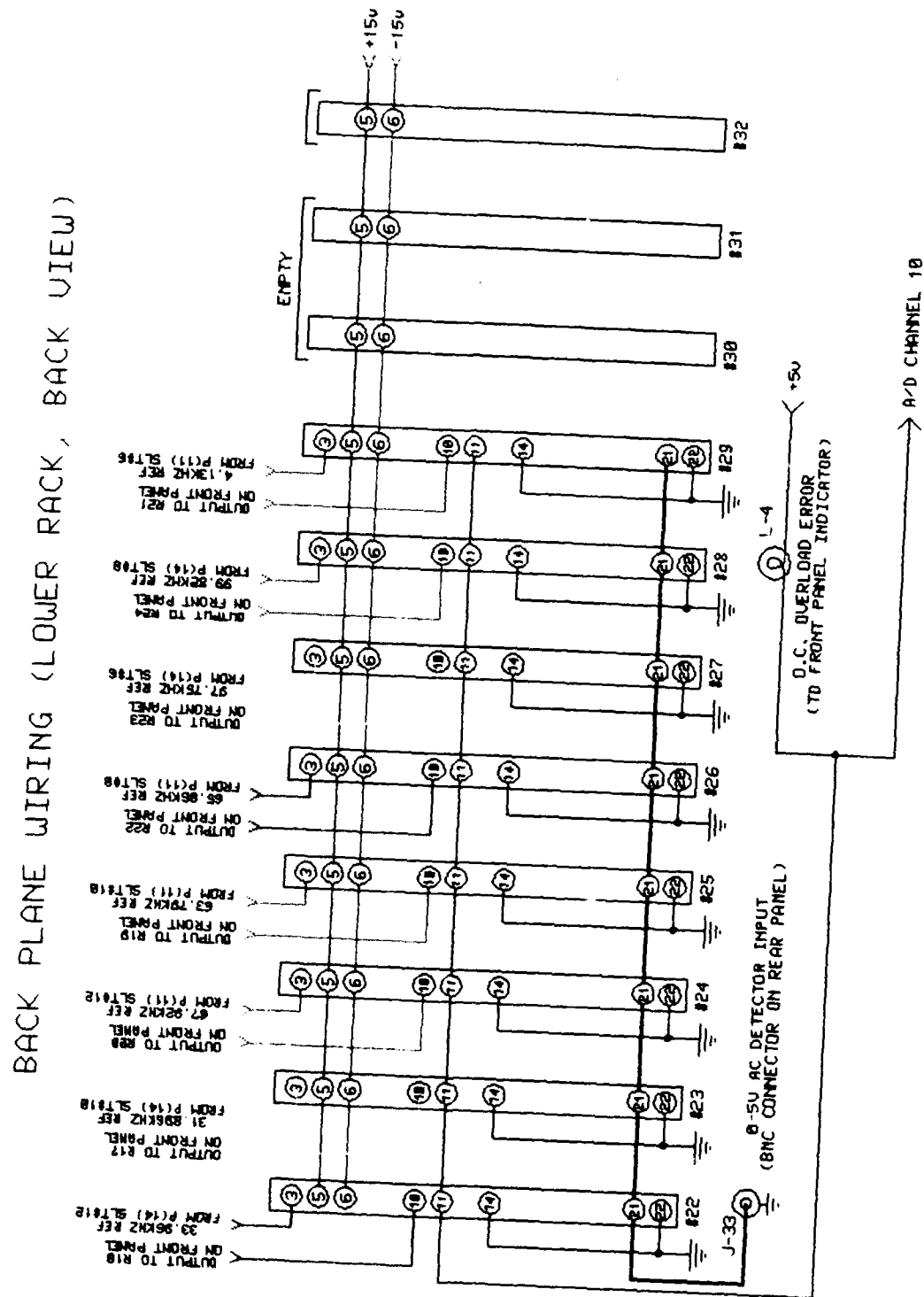
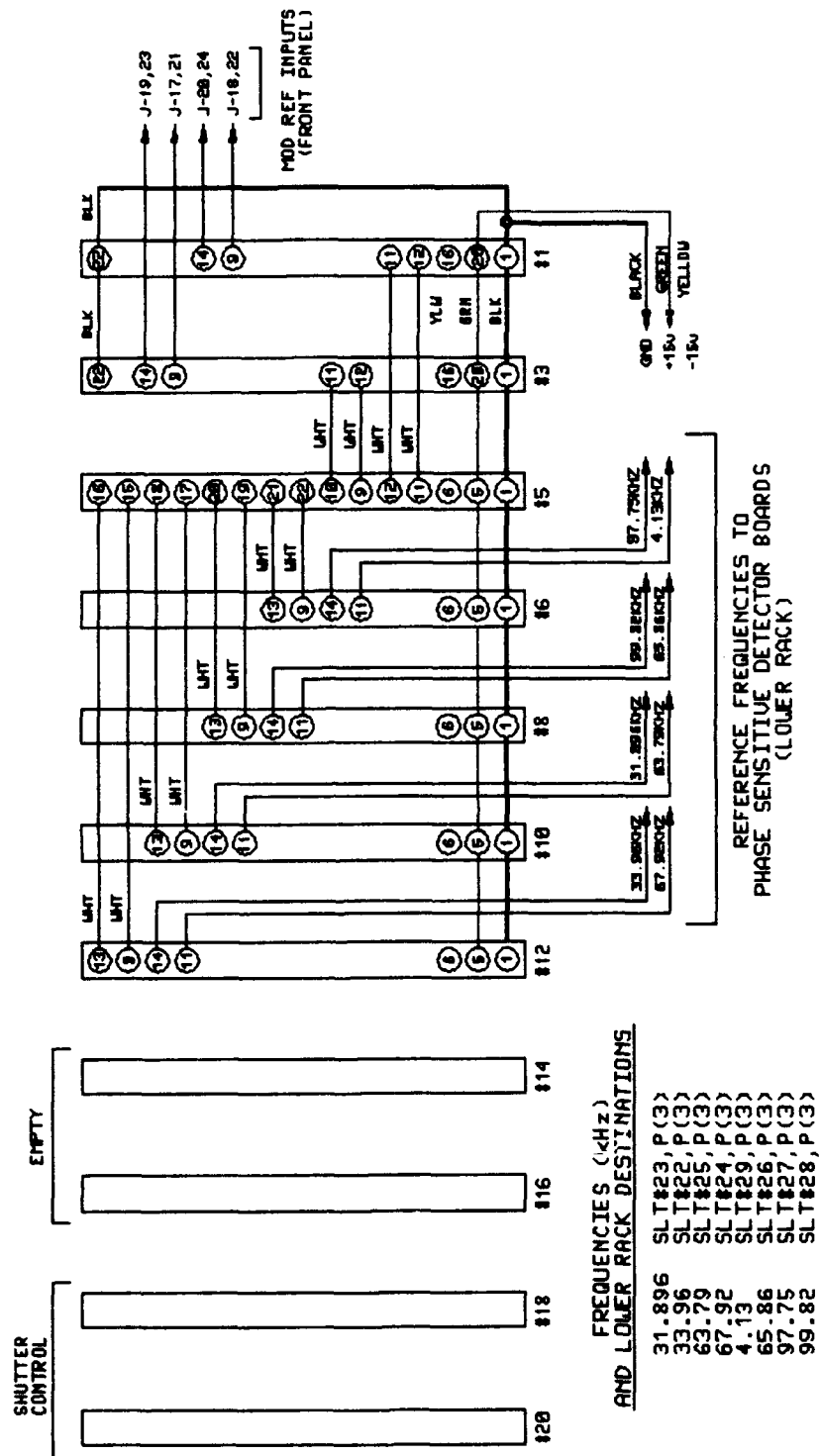


Figure AIII.3. The APSD back panel wiring layout, lower rack.



-125-

# PEM INPUTS CONTROL PANEL / BACK PLANE WIRING

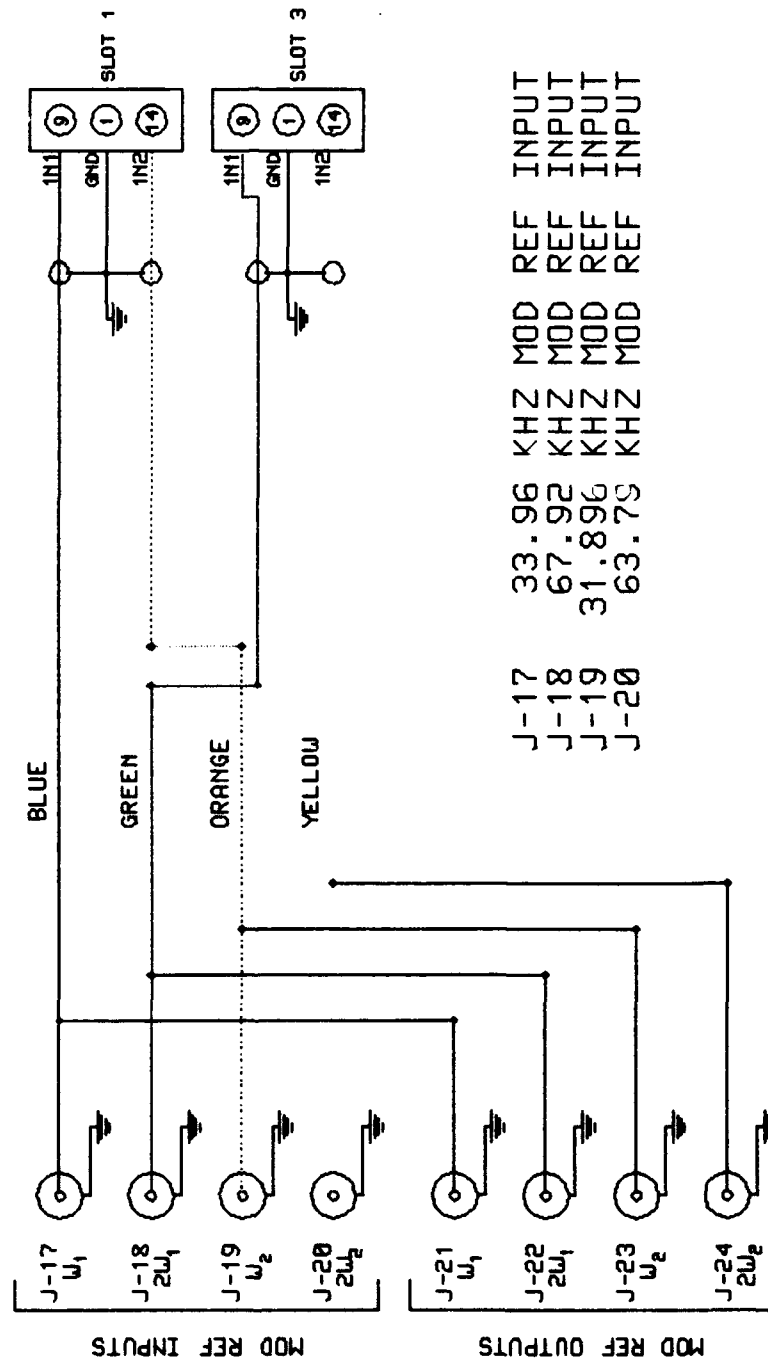


Figure AIII.5. The APSD control to back plane wiring harness. (Reference inputs.)

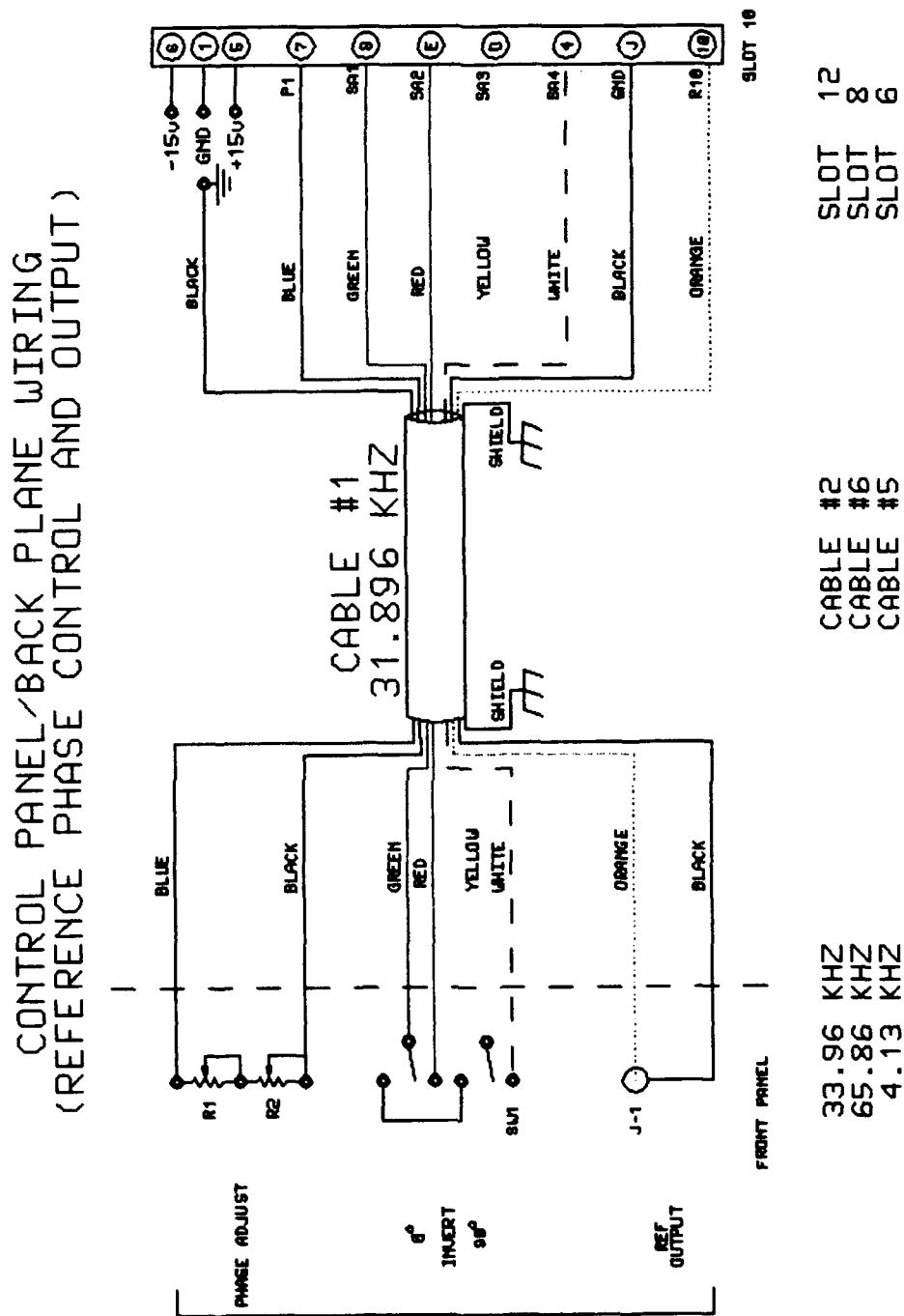


Figure AIII.6a. The APSD control panel to back plane wiring harness. (Reference phase control and output.)

CONTROL PANEL/BACK PLANE WIRING  
(REFERENCE PHASE CONTROL AND OUTPUT)

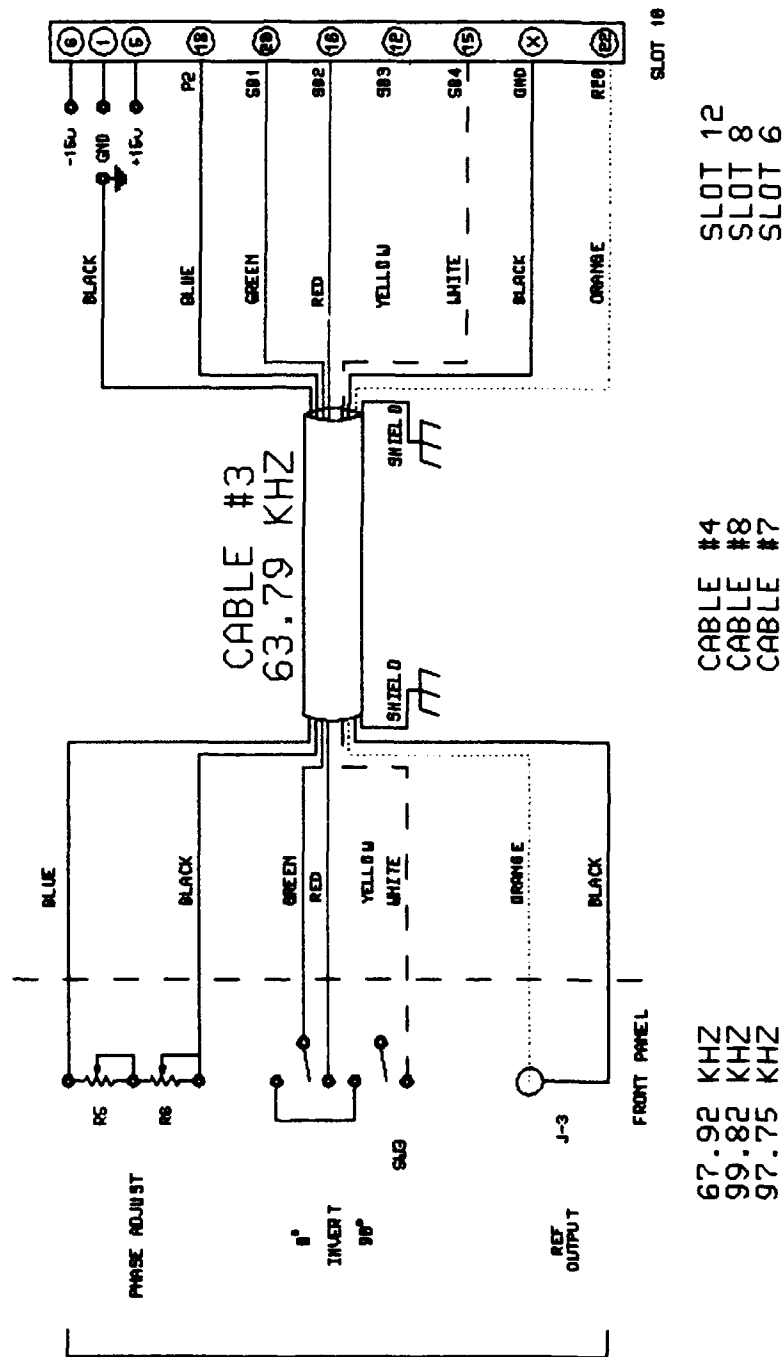


Figure AIII.6b. The APSD control panel to back plane wiring harness. (Reference phase control and output.)

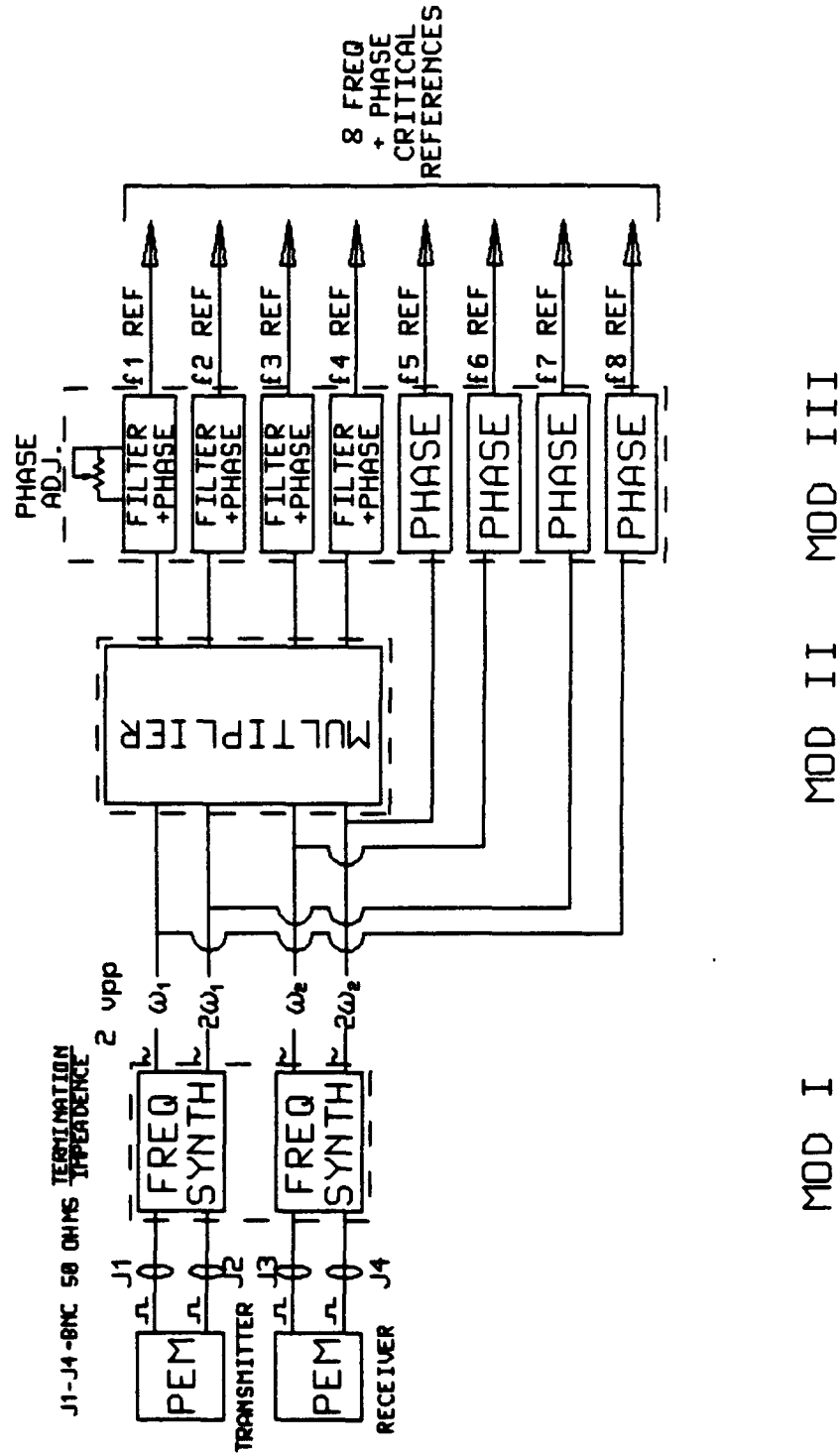


Figure AIII.7a. Flowchart of PEM modulator, frequency synthesizer and multiplier, and phase adjustment circuits of the APSD (Modules I-III).

## ANALOG MULTIPLIER BOARD SCHEMATIC

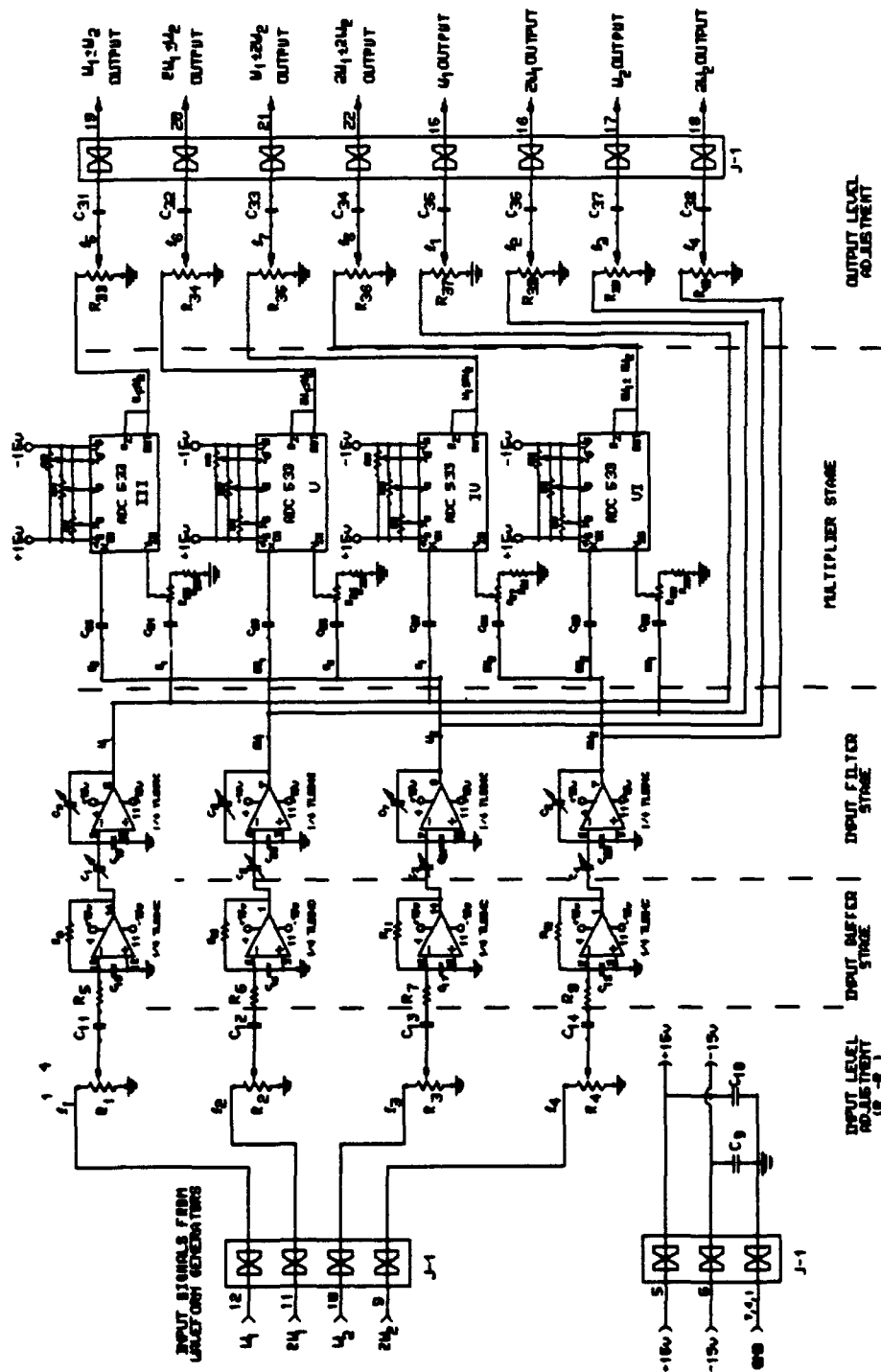
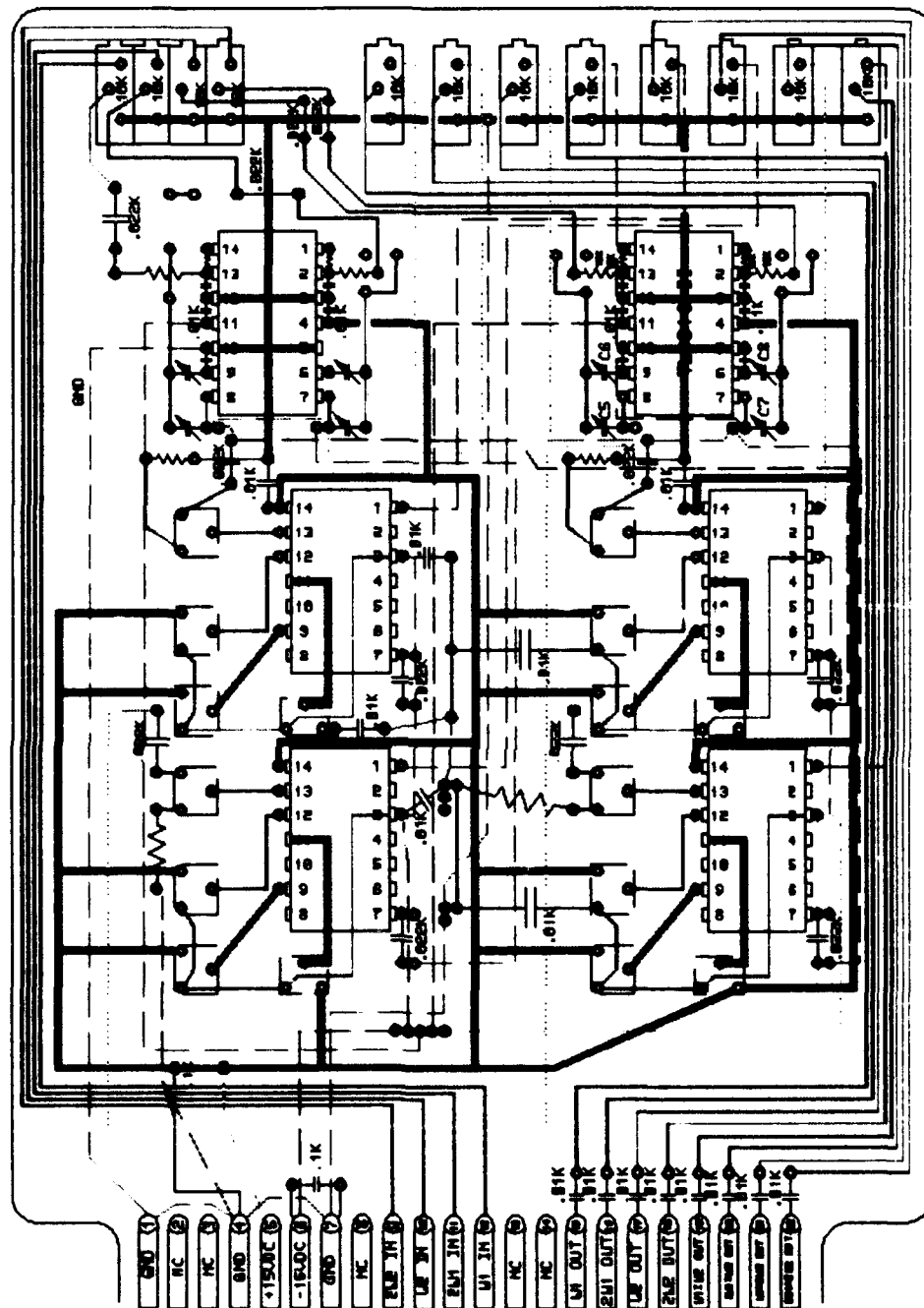


Figure AIII.7b. Schematic of the APSD analog multiplier board (Module II).



-131-

FILTER AND PHASE BOARD LAYOUT (SOLDER SIDE)

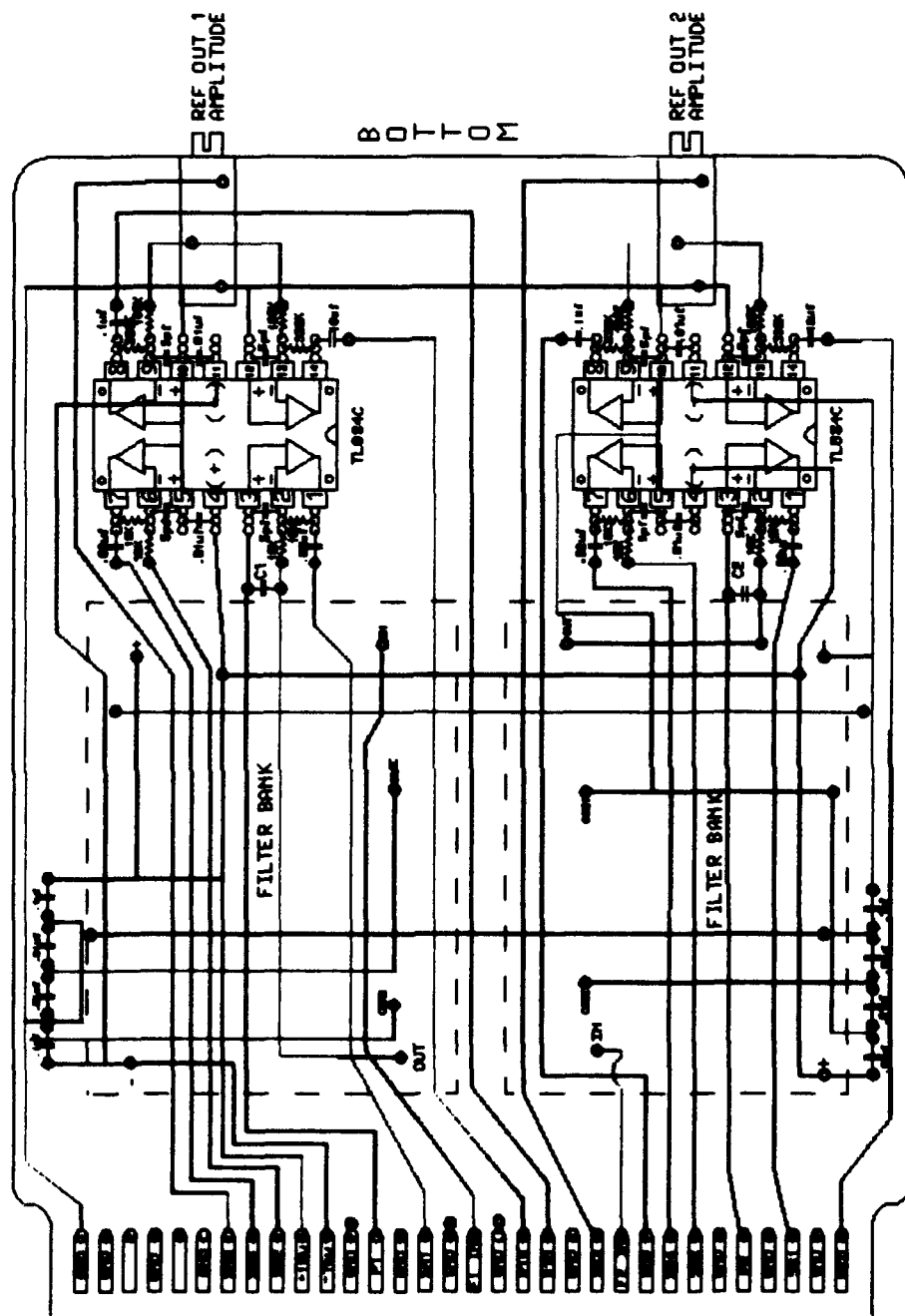
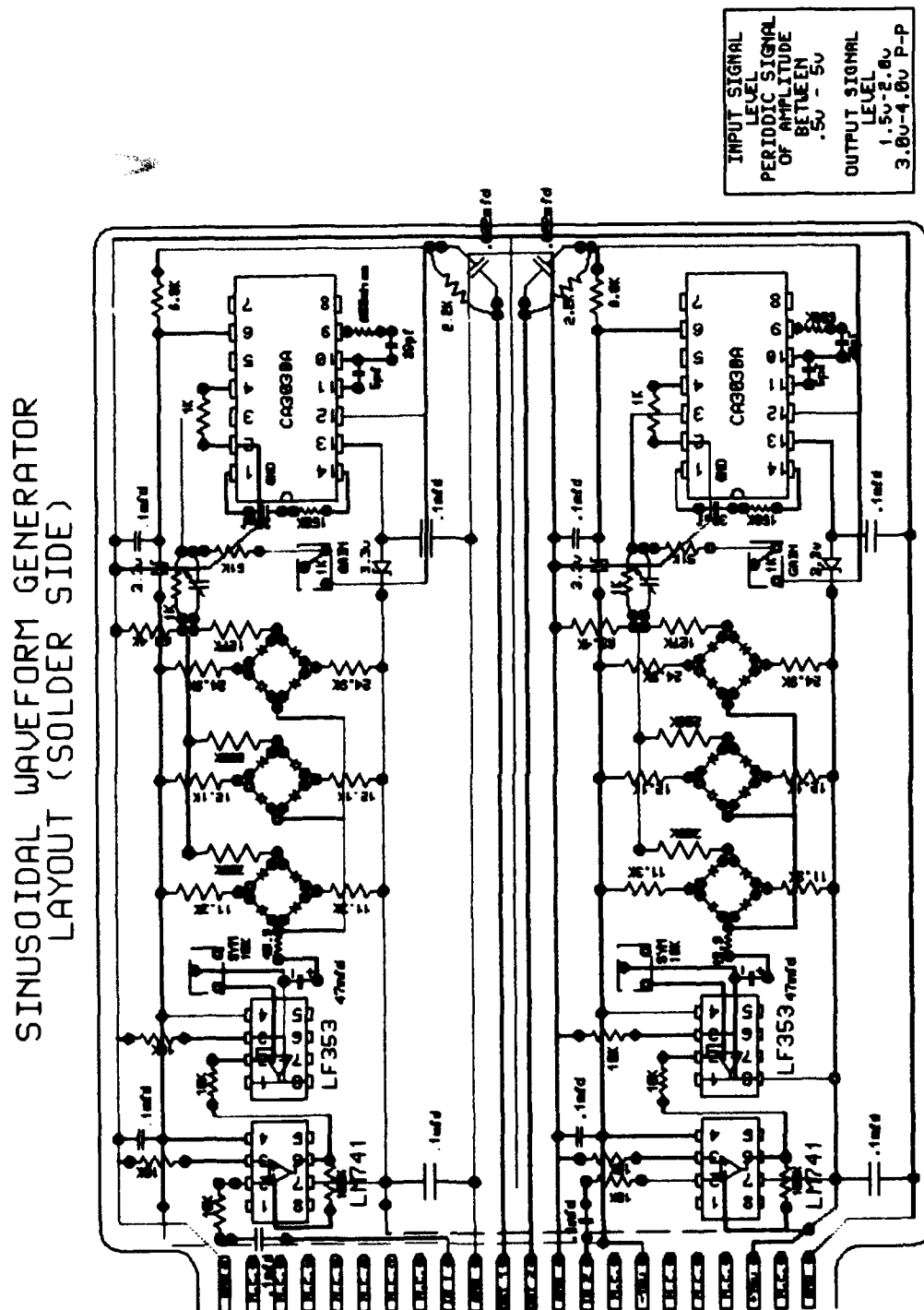


Figure AIII.7c. The APSD filter and phase board layout design (Module III).



**Figure AIII.7d. Schematic of the APSD sinusoid waveform generator (Modules I).**

# SINUSOIDAL WAVEFORM GENERATOR SCHEMATIC

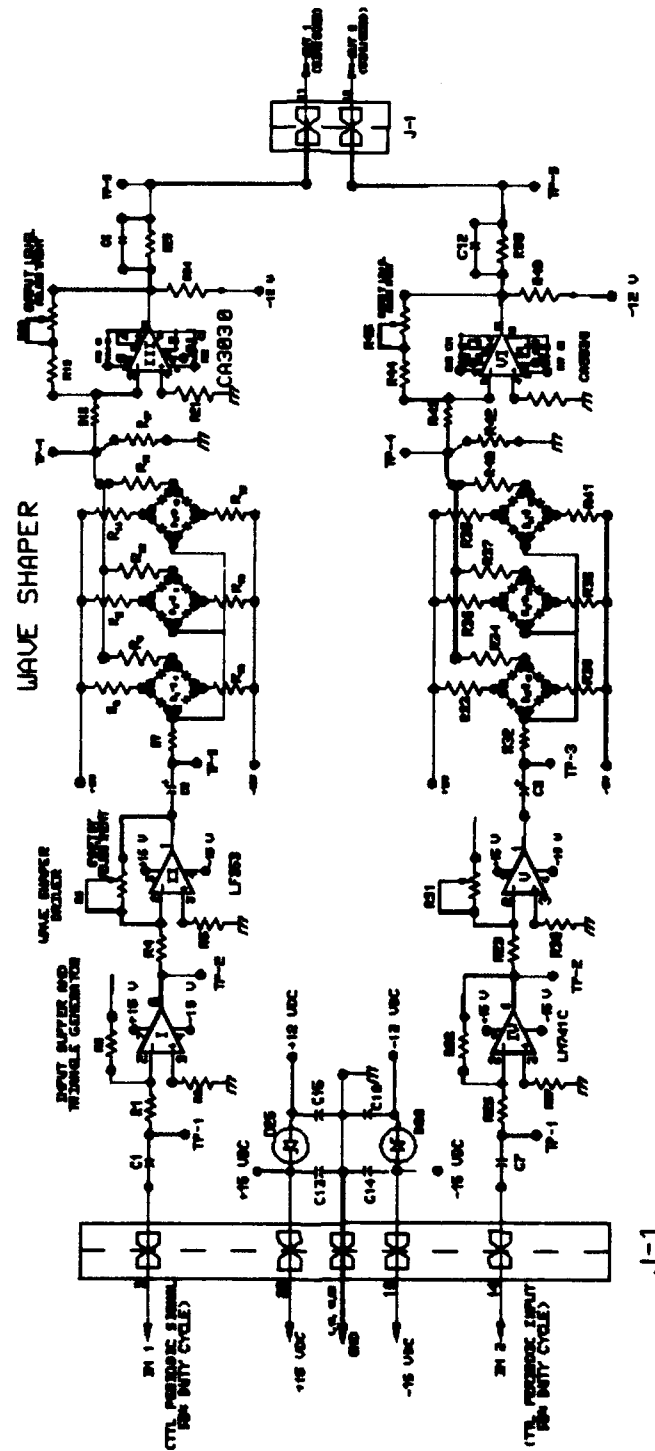


Figure AIII.7e. The APSD sinusoid waveform generator board layout design (Module I).

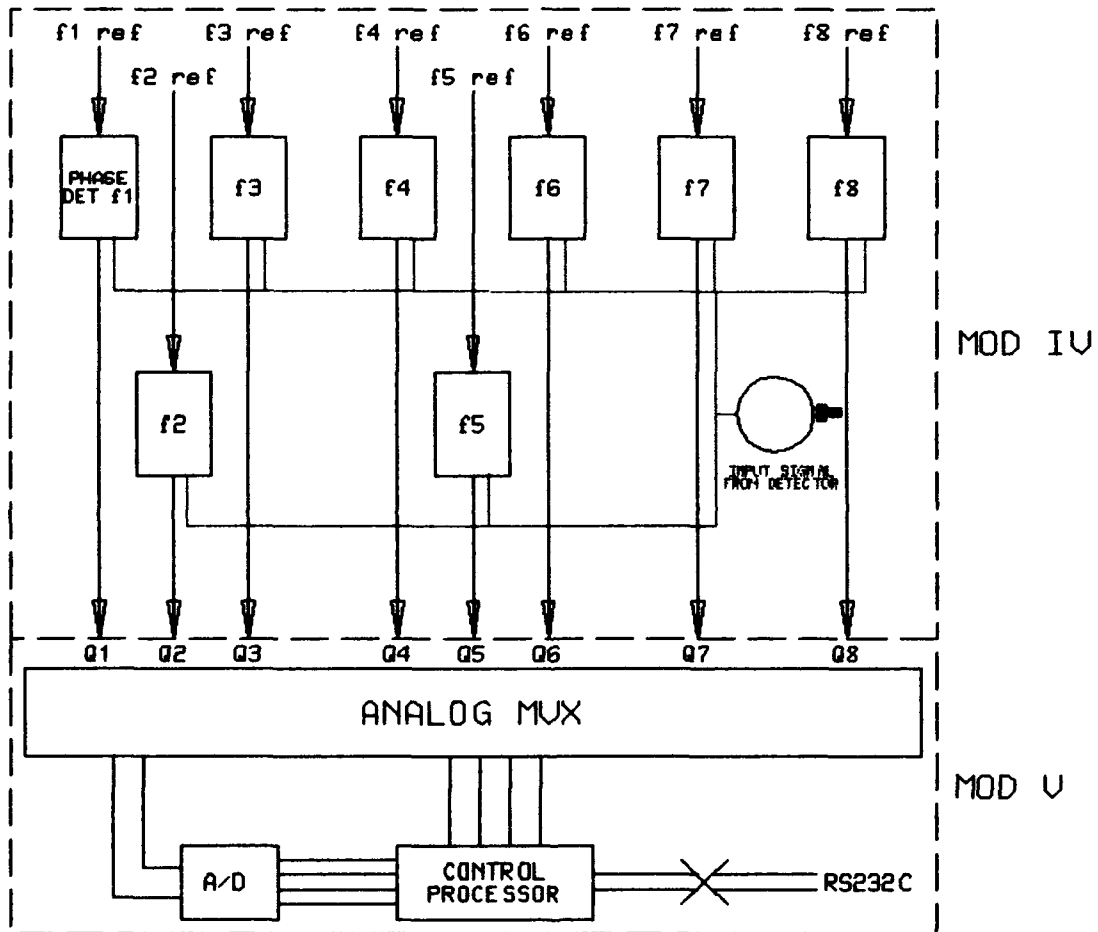


Figure AIII.8. Flowchart of phase references, the Phase Sensitive Detectors, and the DAEDAL ST701 A/D board circuits of the APSD (Modules IV-V).

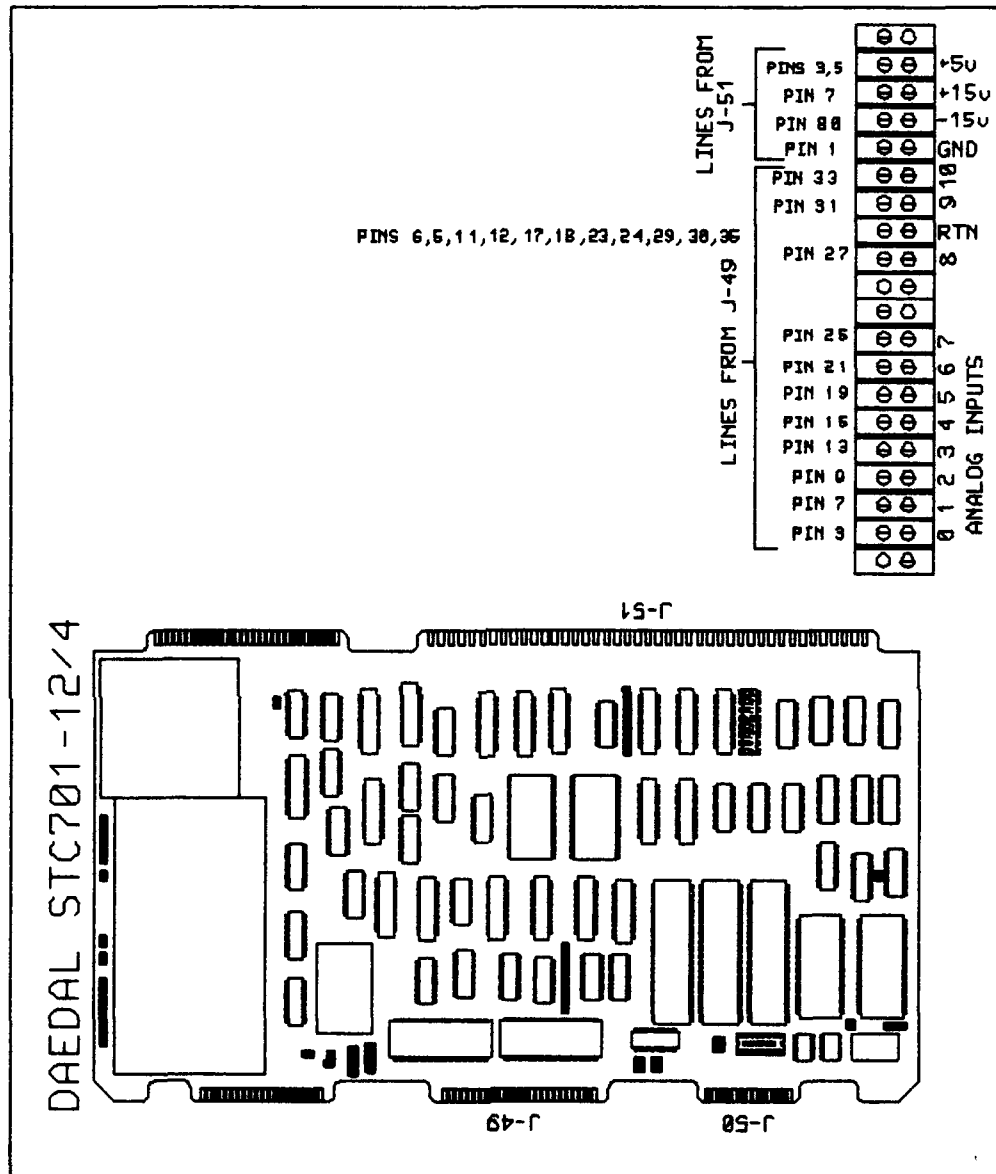


Figure AIII.9a. Module V pin connections to the DAEDAL ST701 A/D microprocessor.

# Appendix III

J-49 PIN #2	
	#1
	#4
CHANNEL 0	#3
RETURN 1	#6
RETURN 0	#5
	#8
CHANNEL 1	#7
	#10
CHANNEL 2	#9
RETURN 3	#12
RETURN 2	#11
	#14
CHANNEL 3	#13
	#16
CHANNEL 4	#15
RETURN 5	#18
RETURN 4	#17
	#20
CHANNEL 5	#19
	#22
CHANNEL 6	#21
RETURN 7	#24
RETURN 6	#23
	#26
CHANNEL 7	#25
	#28
CHANNEL 8	#27
RETURN 9	#30
RETURN 8	#29
	#32
CHANNEL 9	#31
	#34
CHANNEL 10	#33
	#36
RETURN 10	#35
	#38
	#37
	#40
	#39

TO TERMINATION BLOCK

TO J-49 (ON A/D)

NOTE: ALL RETURNS ARE TIED TO A COMMON RETURN ON THE TERMINATION BLOCK.

Figure AIII.9b. Module V wiring assignments of J-49 connector to the DAEDAL ST701 A/D microprocessor.

PHASE SENSITIVE DETECTOR/ A/D CONVERTER  
WIRING DIAGRAM

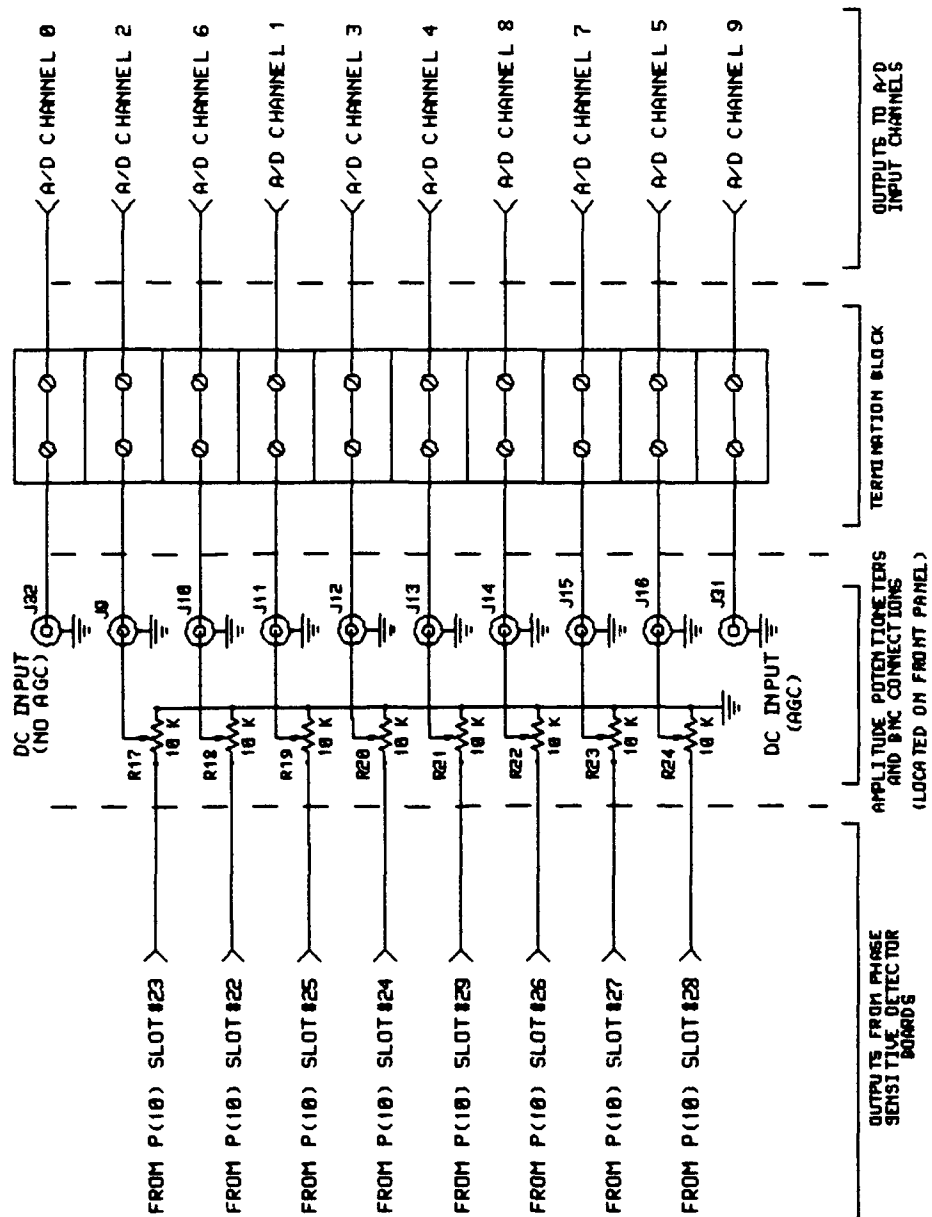


Figure AIII.10. A/D converter (Module V) to Phase Sensitive Detector wiring assignments.

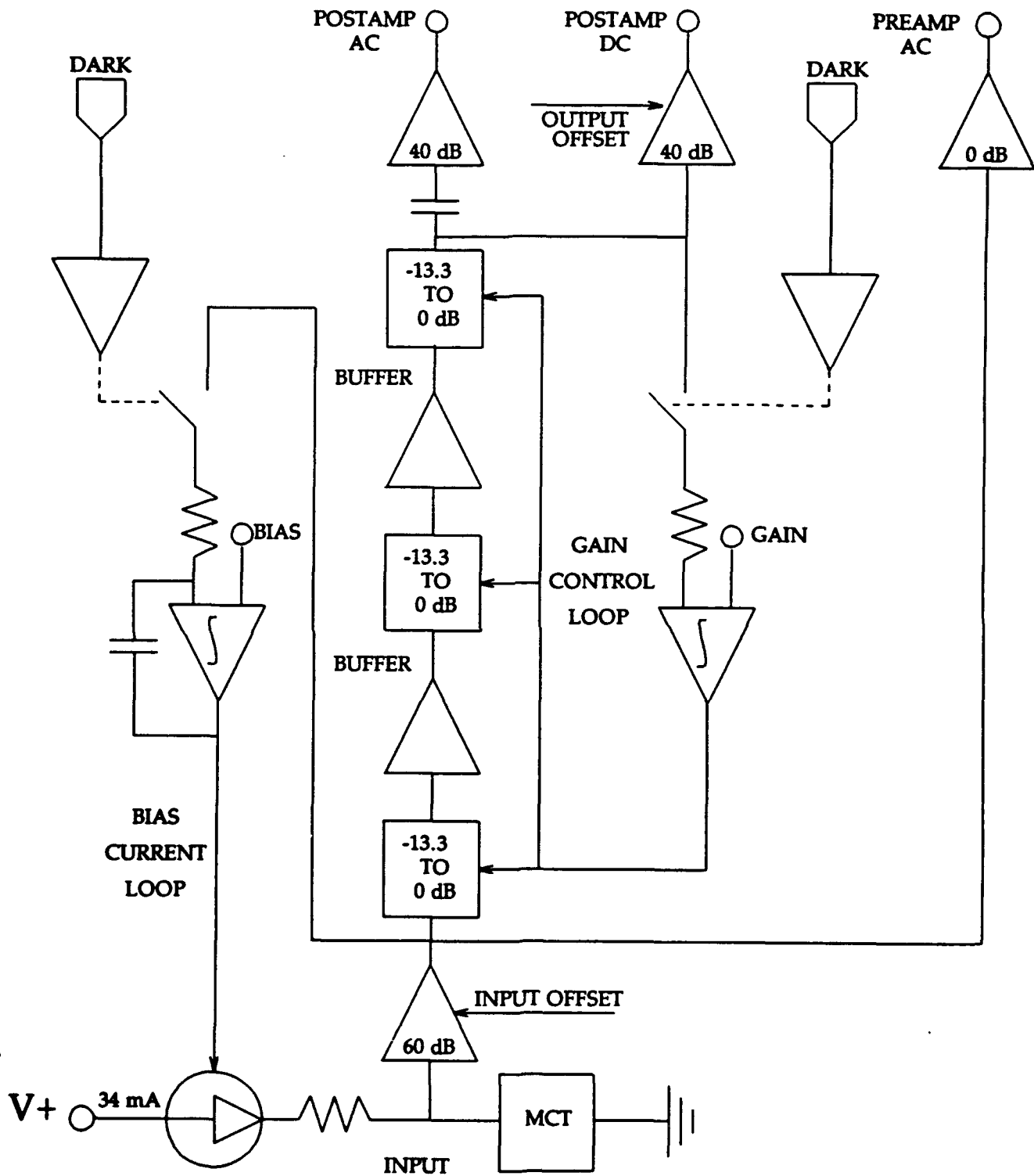


Figure AIII.11. Automatic gain control and amplification circuit (Module VI).

## INCIDENT BEAM POWER REGULATOR SCHEMATIC

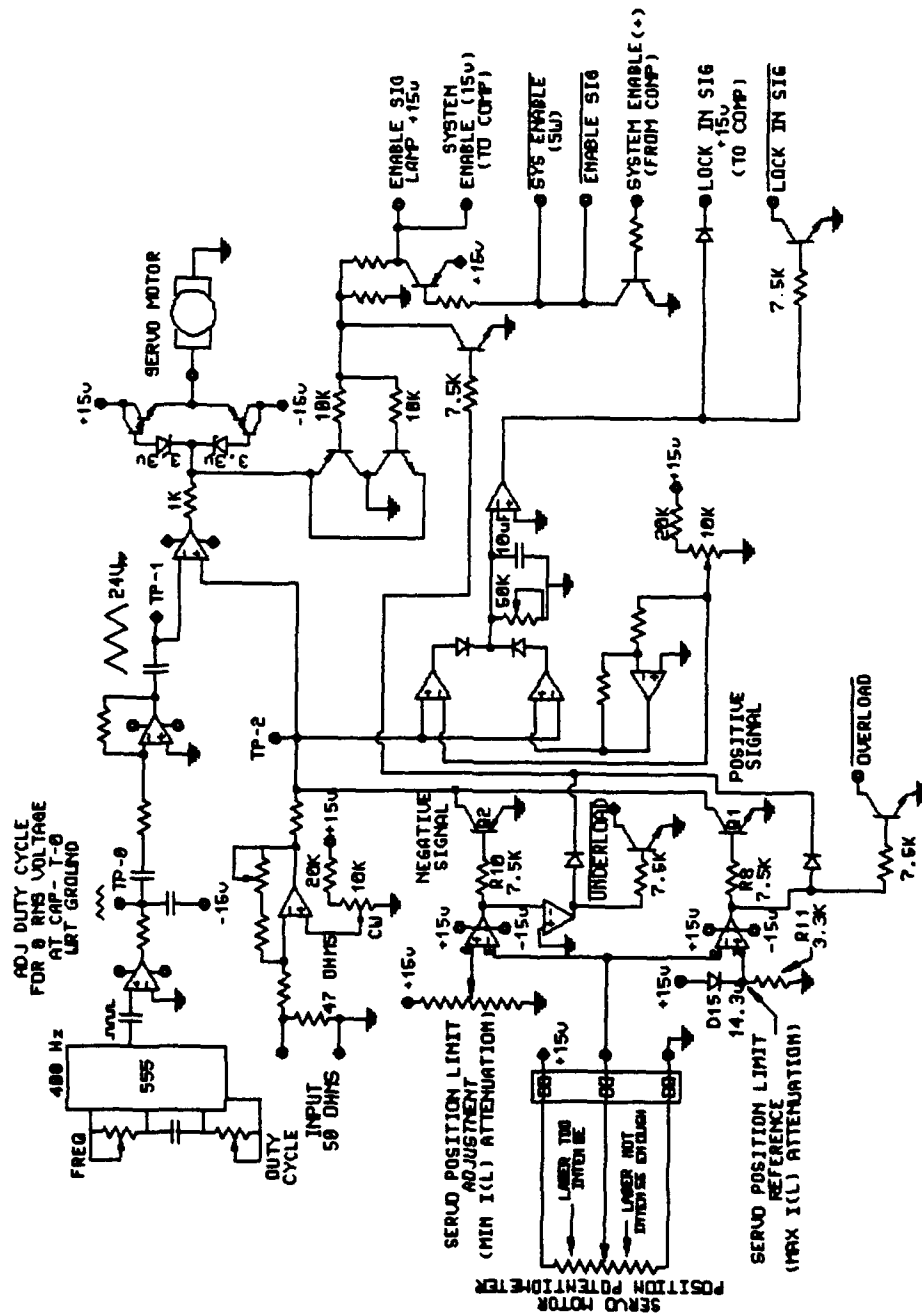


Figure AIII.12a. The incident beam power regulation circuit Module VII. The feedback loop acts between the pyroelectric detector output and the servo motor that adjusts the transmission axis of a polarizing crystal.

# POWER REGULATOR SERVO PLUG, SOCKET, + CORD

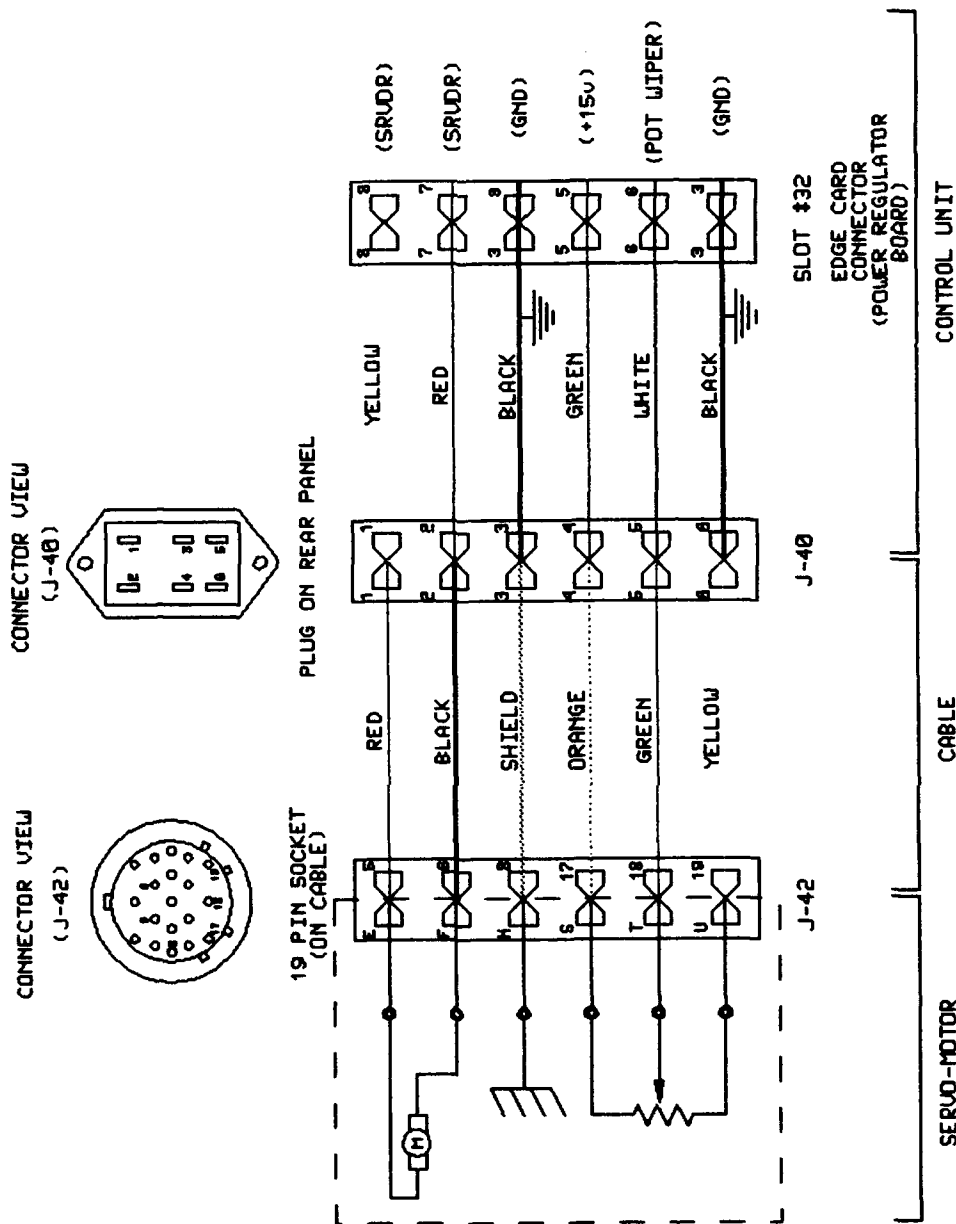


Figure AIII.12b. Layout of the APSD incident beam power regulation servo plug, socket and cord (Module VII).

# Appendix III

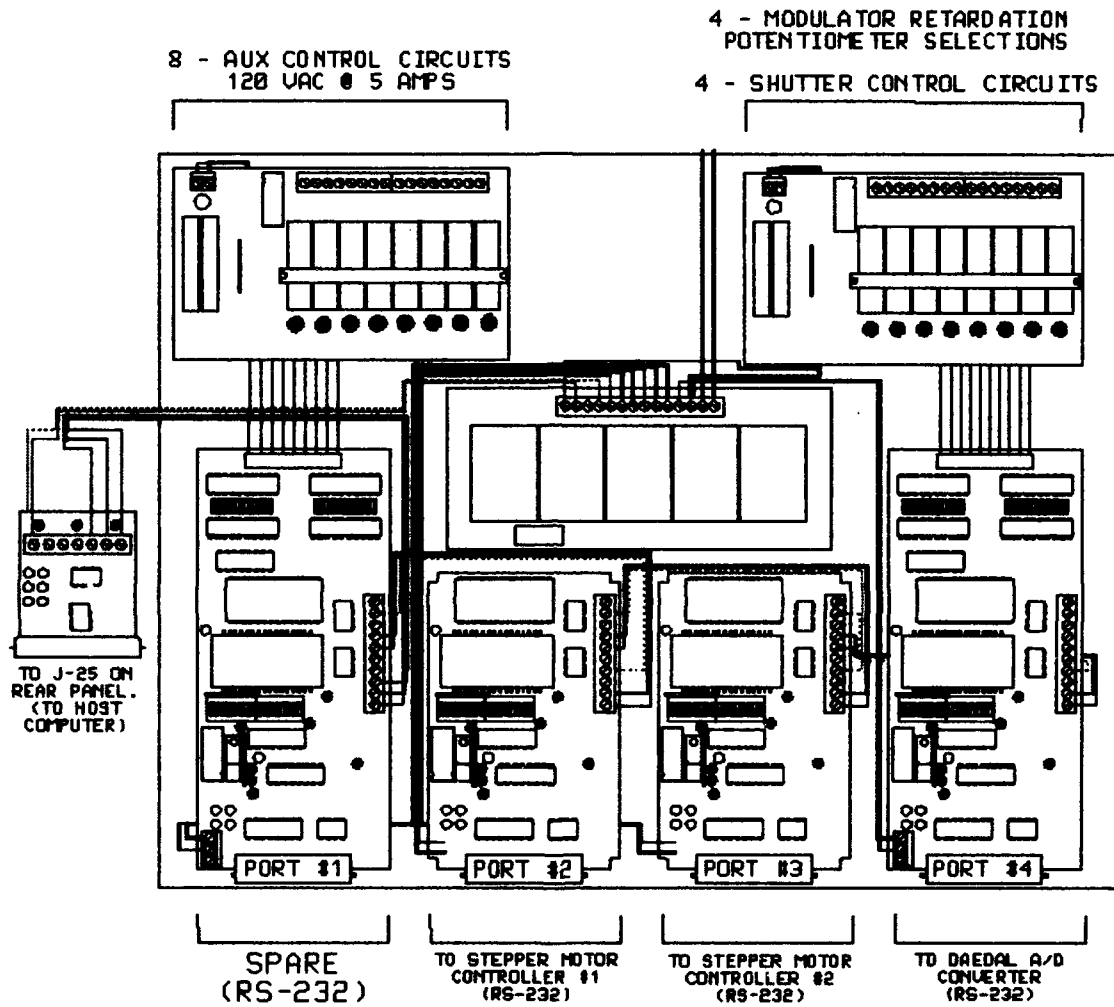


Figure AIII.13a. The APSD Serial Address Gateway (SAG) system (Module VIII).

# SERIAL DATA CABLE TO STEPPER MOTOR CONTROLLER

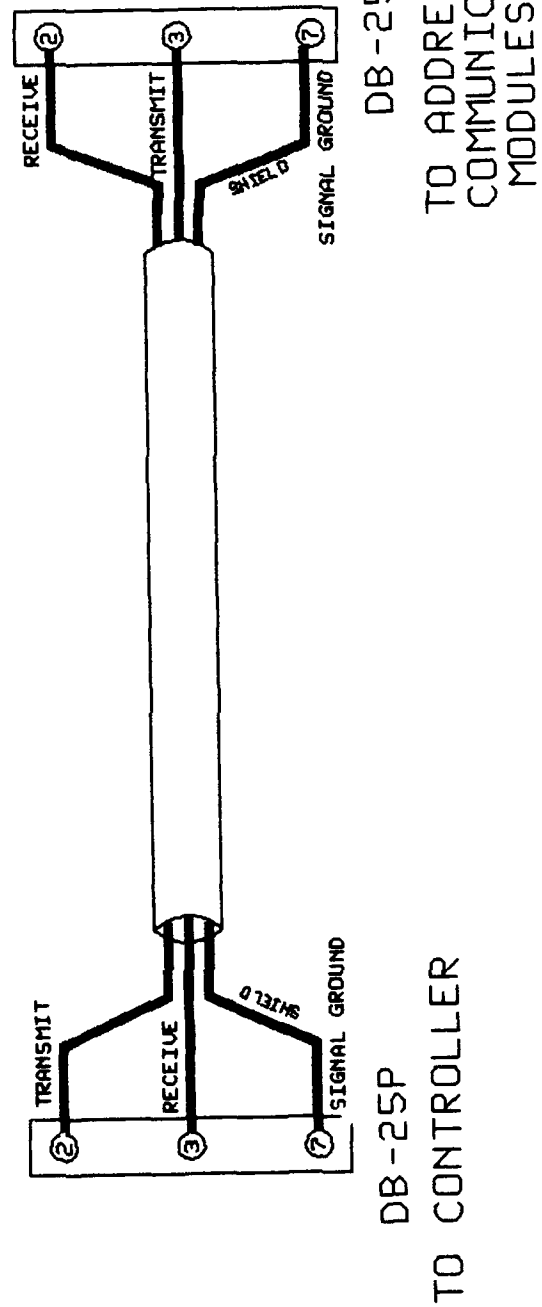


Figure AIII.13b. The serial data cable from the SAG to the stepper motor controller (Module VIII).

# PEM CONTROLLER/APSD WIRING DIAGRAM

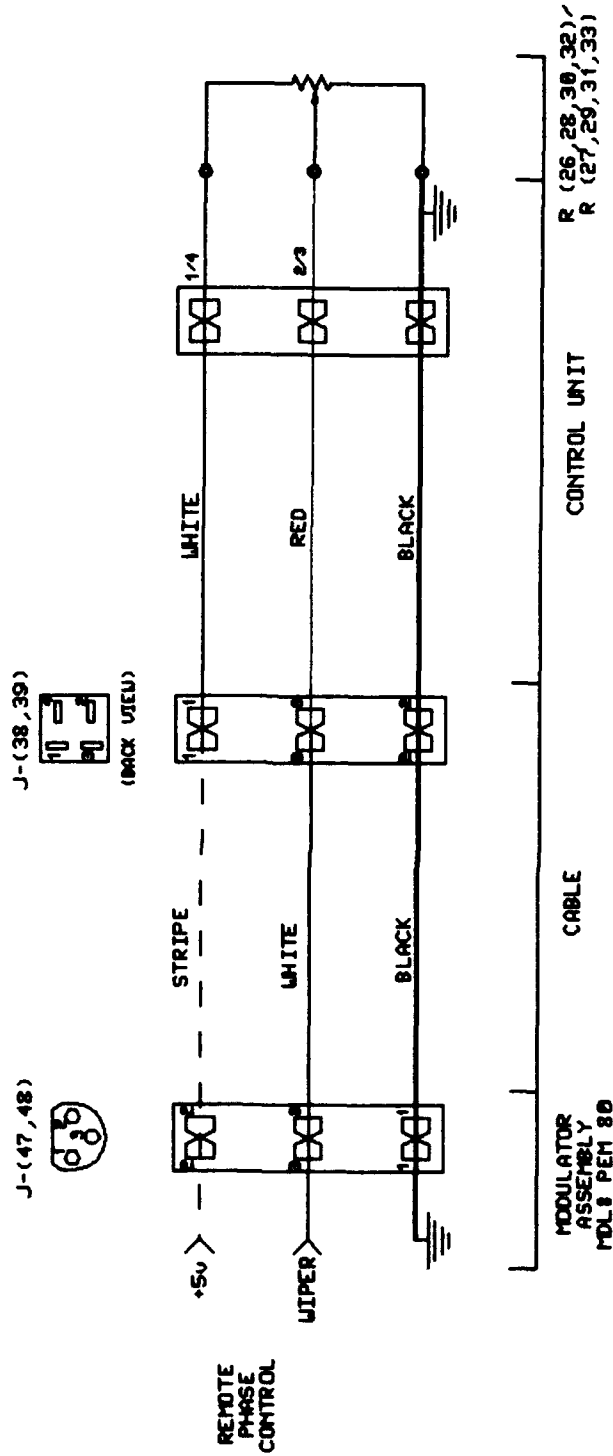
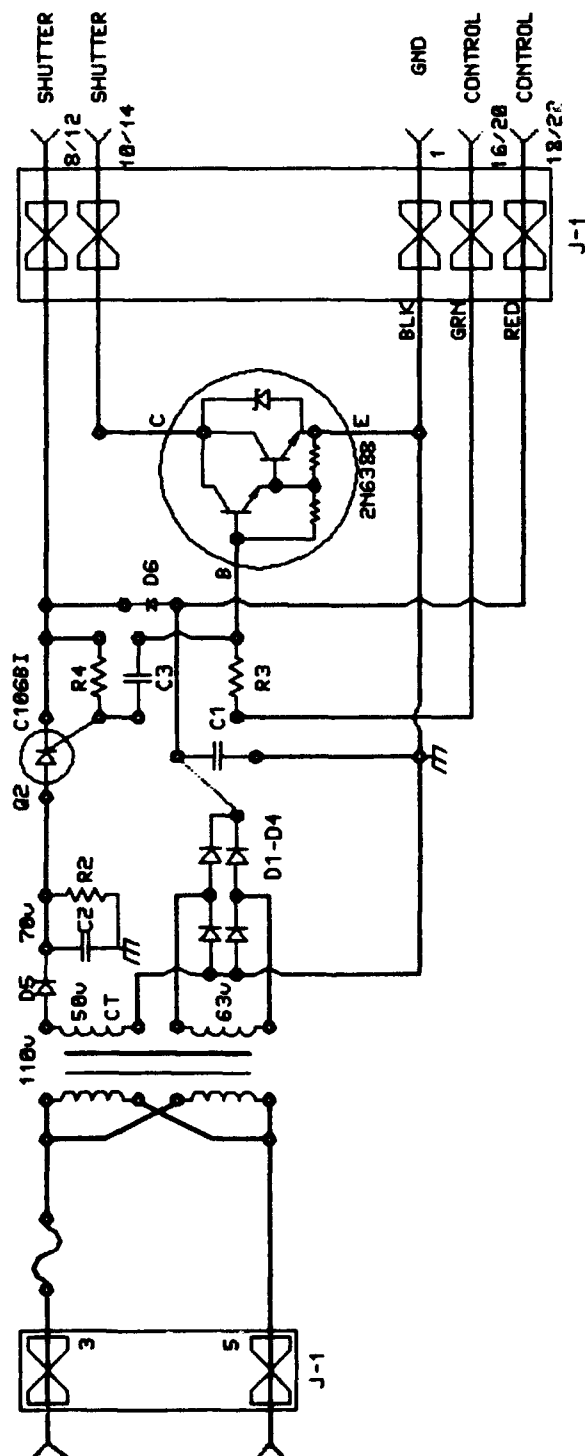


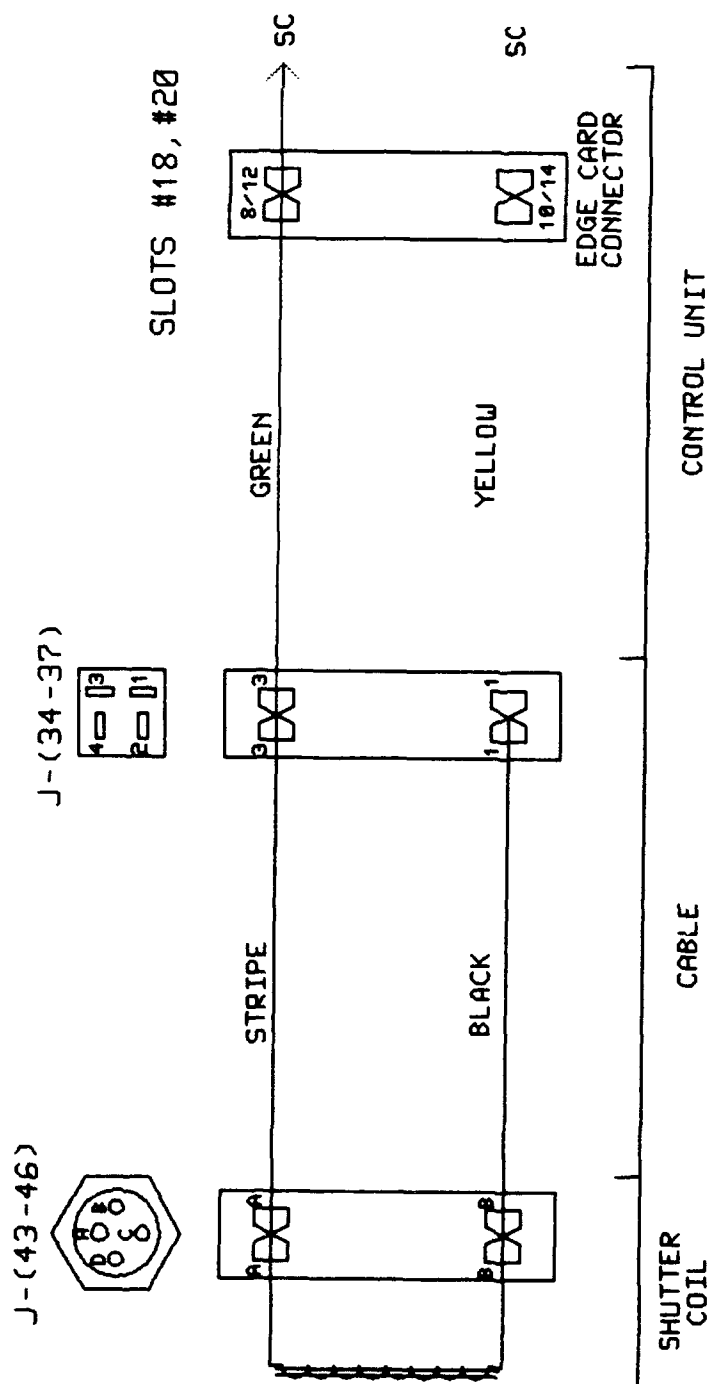
Figure AIII.14. Wiring diagram between transmitter and receiver photoelastic modulators (retardation control) and the SAG.

## SHUTTER CONTROL SCHEMATIC



**Figure AIII.15a.** The APSD shutter control board for switching between incident beams. The shutter drive unit is the model 100-2B UniBlitz design of Vincent Associates, Inc.

## SHUTTER WIRING



**Figure AIII.15b. Wiring diagram between incident beam shutters and the SAG.**

***APPENDIX IV: SOURCE CODE OF THE ANALOG APSD SOFTWARE MODULES.***

#### APPENDIX IV: SOURCE CODE OF THE ANALOG APSD SOFTWARE MODULES.

Program LISA (LIght Scattering Apparatus), written by Charles Henry, consists of 32 FORTRAN 77 subroutine modules. This appendix includes a flowchart of these modules as integrated into the APSD unit, and their source codes.

When entering "RUN MAIN", LISA prompts for five options: (1) begin a new experiment; (2) review collected data; (3) calibrate optics; (4) calibrate A/D converter; and (5) exit. In Step (1), a header block of information is established before the experiment is executed. That information includes (a) the operator name, (b) the number of scattering samples, (c) the sample name, (d) whether the sample is dry or contaminated, (e) the start and final backscattering angle selections, (f) the resolution of backscattering angle scan between these limits, (g) the number of lasers, and (h) the wavelengths of the laser beams. LISA will then ask which communications port with parity, bits, baudrate, and mode is linked to the Serial Addressable Gateway (SAG). LISA then prompts for whether real-time graphics, real-time A/D channel voltages, or neither are to be presented on the screen. (If 'neither' is selected, the experiment will run faster.) After these data are entered, the experiment proceeds automatically. Steps (2) and (5) are self-explicit. The calibration Steps (3)-(4) will typically be performed before and after a long trial of experiments (see Sections 4.6.1 - 4.6.2).

We now present the system flowchart followed by its source code modules (see also Section 4.7).

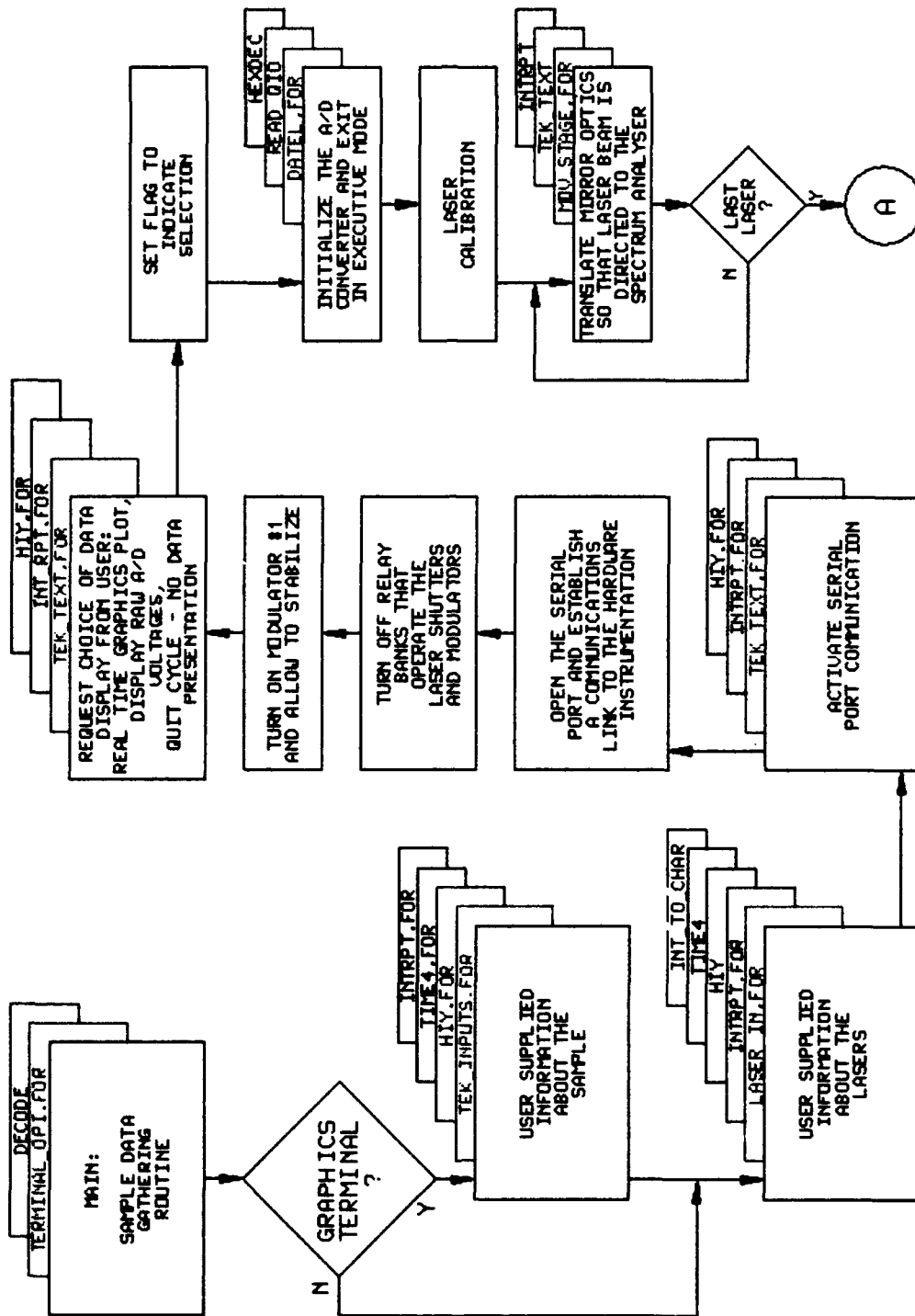
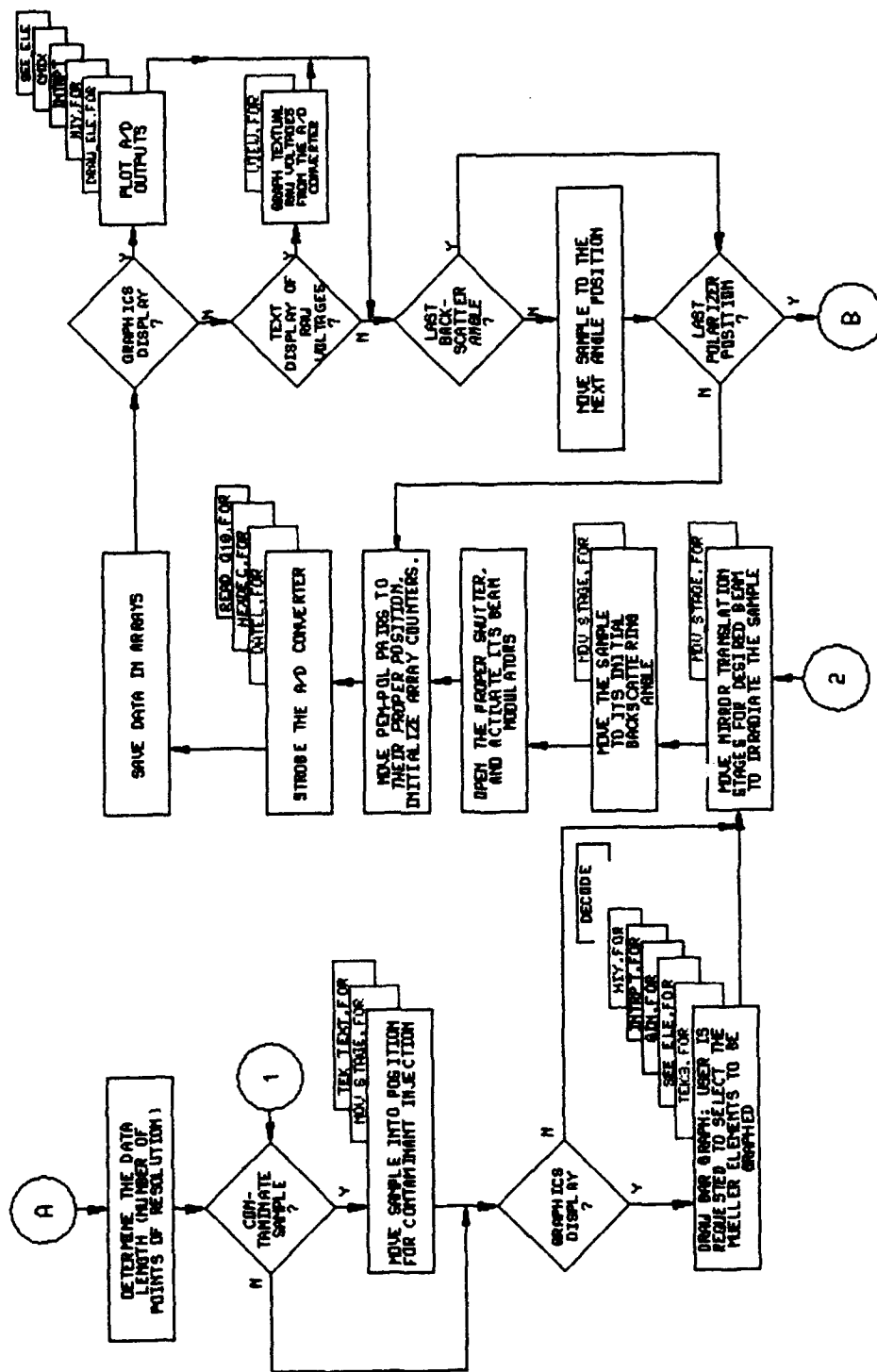


Figure AIV.1 Light Scattering Apparatus (LISA) flowchart of system software modules.



**Figure AIV.1 - continued**

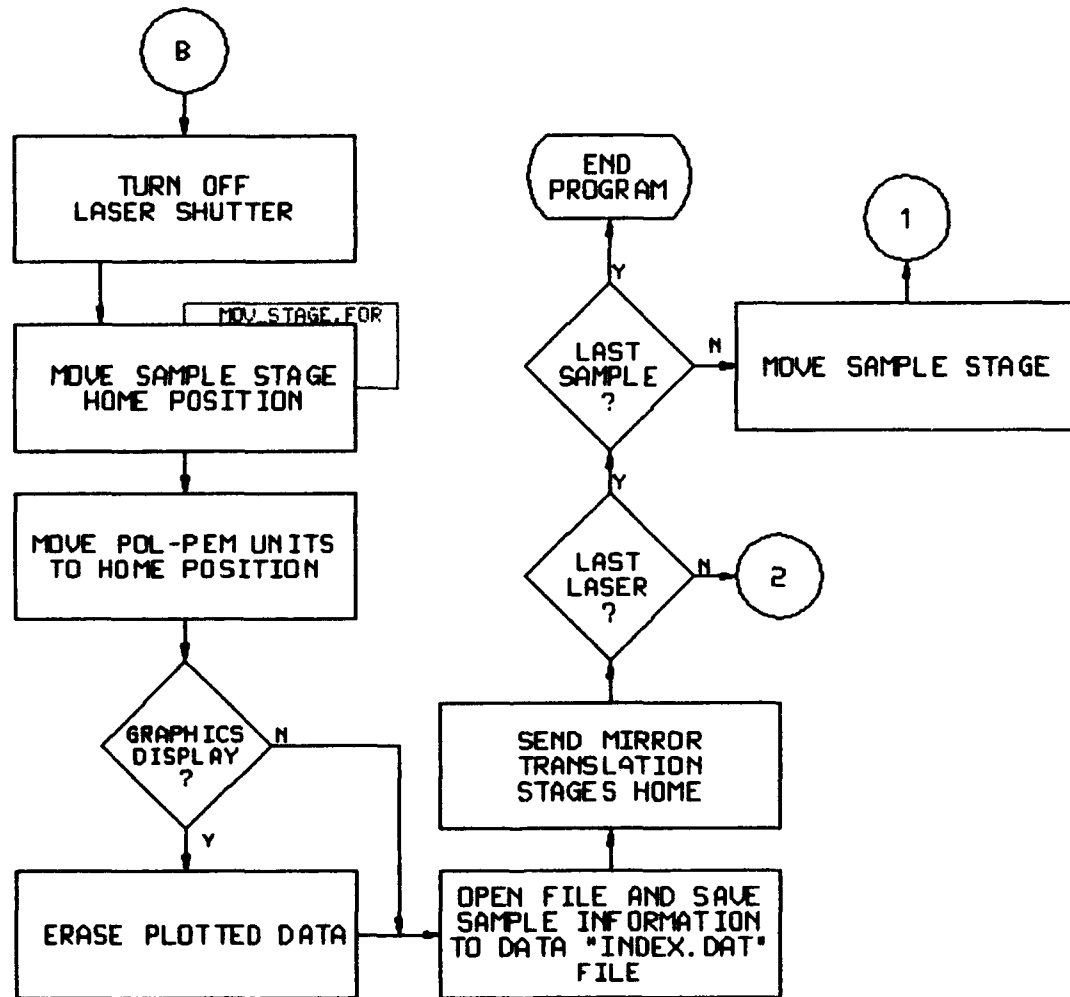


Figure AIV.1 - continued

## Appendix IV

### AIV.1 Analog APSD Software Modules: BUBBLE\_UP Source Code.

```

1      SUBROUTINE BUBBLE_UP(ARRAY,SIZE)
2
3      C-----
4      C   THIS IS A ASCENDING INTEGER BUBBLE SORT ROUTINE. AN ARRAY IS
5      C   PASSED IN THAT IS SORTED FROM 1 - 1000. SIZE IS THE NUMBER
6      C   OF ITEMS IN THE ARRAY THAT THE USER WANTS SORTED.
7      C-----
8
9      INTEGER ARRAY,SIZE,HOLD,BUF
10     DIMENSION ARRAY(*),BUF(20)
11     L = 0
12
13 10    HOLD = 1000
14     J = 0
15
16     DO 20 I = 1,SIZE      ! LOOP THRU THE ARRAY
17         IF(ARRAY(I).GE.HOLD)GOTO 20
18         HOLD = ARRAY(I)    ! HOLD LOWEST NUMBER IN HOLD
19         J = I              ! KEEP PLACE OF LOWEST NUMBER
20 20    CONTINUE            ! END LOOP
21
22     L = L + 1              ! INCREMENT ARRAY COUNTER
23     BUF(L) = HOLD
24     ARRAY(I) = 1000        ! PLACE BIG NUMBER WHERE
25     IF(L.EQ.SIZE)GOTO 30   ! SMALLEST WAS.
26     GOTO 10
27
28 30    DO 40 I = 1,SIZE
29         ARRAY(I) = BUF(I)
30 40    CONTINUE
31
32     RETURN
33     END

```

# Appendix IV

## AIV.2 Analog APSD Software Modules: CMIX Source Code.

```

1      SUBROUTINE CMIX(RNUM,CNUM)
2
3      C-----
4      C   THIS ROUTINE IS DESIGNED TO TAKE IN A REAL NUMBER(UNDER 1000)
5      C   AND CONVERT IT TO A CHARACTER FOR USE WITH A
6      C   GRAPHICS ROUTINE.
7      C-----
8
9      REAL RNUM R1,RPLC,RBUF
10     CHARACTER CNUM*6
11     DIMENSION R1(5),RPLC(4),I1(6)
12     DATA RPLC/1000.,100.,10.,1./
13     CNUM=' '
14 1    FORMAT(F6.2)
15
16     RBUF = RNUM
17
18     DO 10 I=1,4
19     IF(RNUM.GE.RPLC(I))THEN
20     R1(I)=RNUM/RPLC(I)
21     I1(I)=IFIX(R1(I))
22     RNUM=RNUM-(FLOAT(I1(I))*RPLC(I))
23
24     ELSE
25     I1(I)=0
26     ENDIF
27 10    CONTINUE
28
29     IF(RNUM.GE..1)THEN
30     R1(5)=RNUM*.1
31     I1(6)=IFIX(R1(5)*10)
32     I1(6)=I1(6)/10
33     ELSE
34     I1(5)=0
35     ENDIF
36
37     CNUM(1:1)=CHAR(I1(1)+48)
38     CNUM(2:2)=CHAR(I1(2)+48)
39     CNUM(3:3)=CHAR(I1(3)+48)
40     CNUM(4:4)=CHAR(I1(4)+48)
41     CNUM(5:5)=CHAR(I1(5)+48)
42     CNUM(6:6)=CHAR(I1(6)+48)
43
44     DO 20 I=1,6
45     IF(CNUM(I:I).EQ.CHAR(48))THEN
46     CNUM(I:I)= ' '
47     ELSE
48     GOTO 30
49     ENDIF
50 20    CONTINUE
51

```

## Appendix IV

```
52      30  RNUM - RBUF  
53  
54      RETURN  
55      END
```

## Appendix IV

### *AIV.3 Analog APSD Software Modules: DATEL\_1 Source Code.*

```

1      SUBROUTINE DATEL(TYPE,REPORT,PORT,ICNT,CHAN,RDAT,HEX)
2
3
4      C-----
5      C
6      C   This mod is designed to initiate the executive level control
7      C   commands for the date1 701 A/D converter. The first part wakes
8      C   up the system with a carriage control. The executive is accessed
9      C   by the turnon command (line 10). The converter will respond with
10     C   an echo of the meag and provide a carriage control LP and give an
11     C   ~ as a prompt (character string 42). At this point
12     C   the system is ready to collect data.
13     C
14     C   At this point the converter echoes the command and initiates a LP
15     C   CR which I look for and omit from the data collection. The data
16     C   values come in a long string separated by LP CR. I look for these
17     C   and it is used to shape the data so that individual numbers can
18     C   be placed in an array called " RDAT ".
19     C
20     C
21     C   TYPE - THE FUNCTION TO BE PERFORMED BY THIS ROUTINE
22     C           0. - INITIALIZE AND STROBE 9 CHANNELS
23     C           1. - INITIALIZE THE A/D TO EXECUTIVE MODE ONLY
24     C           2. - STROBE ONLY ONE CHANNEL PASSED IN BY " CHAN "
25     C           3. - STROBE CHANNELS 0 - 16 BASED ON ICNT
26     C
27     C   REPORT - ERROR FLAG WHEN SET TO 1 INDICATES THAT THE ROUTINE
28     C           DID NOT FUNCTION CORRECTLY. THIS IS PASSED BACK TO
29     C           THE CALLER
30     C
31     C   PORT - THIS IS THE SERIAL PORT DEFINED BY THE CALLER THAT
32     C           WILL BE USED FOR THE A/D COMMUNICATIONS.
33     C
34     C   ICNT - THE NUMBER OF DATA ELEMENTS ( OR CHANNEL NUMBERS )
35     C           THAT DATA IS BEING COLLECTED FOR. THIS IS REQUIRED
36     C           BY THE HEXDEC SUBROUTINE.
37     C
38     C   CHAN - THE INDIVIDUAL CHANNEL NUMBER TO BE STROBED. PASSED
39     C           IN FROM THE CALLER
40     C
41     C   RDAT - ARRAY OF REAL NUMBERS CONVERTED FROM HEXDECIMAL
42     C           DEFINING THE RELATIVE VOLTAGES FOR EACH CHANNEL.
43     C           THIS IS PASSED BACK TO THE CALLER.
44     C-----
45
46     CHARACTER MSG,A*80,TDAT*1,ENDDAT*5,TURNON*19,CS,CR,LP
47     CHARACTER MSG1*30,PORT*10,MSG*512,CDAT*4,QUIET*3
48     CHARACTER STROBE*9,CHAN*2,CNT*9,HEX*4,A2D_INPUT*6,E*1
49
50     INTEGER*4 TIMEOUT,LEN_STRING
51     INTEGER ISIZE,IOUT,CHANNEL,TYPE,REPORT

```

# Appendix IV

```

52
53     REAL RDAT
54
55     DIMENSION RDAT(16),HEX(16),A2D_INPUT(16)
56
57
58     L   - 0
59     E   - CHAR(27)
60     GS  - CHAR(29)
61     CR  - CHAR(13)
62     LF  - CHAR(10)
63     ICNT - 10
64     IERR - 0
65
66     TIMEOUT - 1
67     IRESCAN - 0
68     TURNON - 'ST-701 EXECUTIVE ON'
69     LEN_STRING - 512
70
71 C   OPEN('4,FILE='WA3',STATUS='NEW',CARRIAGECONTROL = 'NONE')
72
73 C-----
74
75     IF(TYPE.EQ.3.OR.TYPE.EQ.2)GOTO 40    ! STROBING FOR DATA
76     IF(TYPE.EQ.1.OR.TYPE.EQ.0)THEN ! USERS IS INITIALIZING THE BOARD
77
78 C-----
79
80 2   FORMAT(A1)
81
82 10  FORMAT (A30)
83
84     WRITE(3, '(A4)'GS//DO '    ! CLOSE EXISTING COMMUNICATIONS
85     CALL TWAIT(2)           ! PAUSE .2 SECS
86     WRITE(3, '(A4)'GS//D4//CR ! WAKE THE CONTROLLER BOARD
87
88     CALL TWAIT(2)           ! PAUSE .2 SECONDS
89
90     IERR - 0                ! INITIALIZE ERROR COUNT
91
92 16  WRITE(3, '(A1)'CR        ! CARRIAGE CONTROL
93     CALL TWAIT(2)           ! PAUSE .2 SECONDS
94
95     CALL READ_QIO (PORT,MESG,LEN_STRING,TIMEOUT,ISIZE,IOUT,
96     .CHANNEL,INUSE)
97
98
99
100 C   WRITE(4,*)ISIZE, ' INITIALIZING MESG - ',MESG
101
102
103
104
105     DO 17 I = 1,ISIZE + 5    ! CHECK FOR A PROMPT

```

# Appendix IV

```

106      IF(MESG(EI).EQ.'-')GOTO 20 ! COMMAND MODE
107      IF(MESG(EI).EQ.'#')GOTO 20 ! COMMAND MODE
108      IF(MESG(EI).EQ.'~')GOTO 30 ! EXECUTIVE MODE
109 17  CONTINUE
110
111      IERR = IERR + 1      ! INCREMENT ERROR COUNTER
112
113      IF(IERR.EQ.5)THEN    ! NOT COMMUNICATING WITH A/D
114          REPORT = 1      ! SET ERROR FLAG
115          GOTO 1000        ! RETURN TO CALLER
116      ENDIF
117
118      GOTO 16              ! GO TRY TO WAKE IT AGAIN
119
120  C-----
121  C  THIS WRITES THE MESSAGE "ST701-EXECUTIVE ON" TO THE BOARD TO SET
122  C  THE MODE TO EXECUTIVE. WHICH THE BOARD WILL ANSWER WITH A " "
123  C-----
124
125 20  IERR = 0
126
127 25  DO 27 I = 1,19
128      WRITE(3,'(A)')TURNON(EI)
129 27  CONTINUE
130
131      WRITE(3,'(A)')CR
132
133      CALL TWAIT(2)        ! PAUSE 1/2 SECOND
134
135      CALL READ_QIO (PORT,MESG,LEN_STRING,ITIME,ISIZE,IOUT,
136      .CHANNEL,INUSE)
137
138      IF(IOUT.EQ.0)GOTO 30 ! THE A/D BOARD ANSWERED MOVE ON
139
140      IERR = IERR + 1      ! INCREMENT ERROR COUNTER
141
142      IF(IERR.EQ.5)THEN    ! NOT COMMUNICATING WITH A/D
143          REPORT = 1      ! SET ERROR FLAG
144          GOTO 1000        ! RETURN TO CALLER
145      ENDIF
146
147      GOTO 25
148
149  C-----
150
151 30  L = 0
152 38  IF(TYPE.EQ.1)GOTO 1000 ! INITIALIZATION COMPLETE
153      ! RETURN TO CALLER
154      ENDIF
155
156  C-----
157  C  THIS PART ASKS THE A/D TO STROBE ITS CHANNELS
158  C-----
159

```

# Appendix IV

```

160 40  WRITE(3, '(A4)') GS//DO '      ! CLOSE EXISTING COMMUNICATIONS
161
162      CALL TWAIT(2)                ! PAUSE .2 SECS
163
164      WRITE(3, '(A4)') GS//D4//CR   ! CONTROL IS TO A/D BOARD
165
166      CALL TWAIT(2)                ! PAUSE .2 SECONDS
167
168  C -----
169  C   THIS REMOVES THE MOTOR CONTROLS FROM THE READ BUFFER
170  C -----
171
172
173      CALL READ_QIO (PORT, MSG, LEN_STRING, ITIME, ISIZE, IOUT,
174      .CHANNEL, INUSE)
175
176  C   WRITE(4, *) ISIZE - ', ISIZE
177  C   WRITE(4, *) MSG - ', MSG      ! TEST DATA
178
179      IERR = 0
180
181 42  WRITE(3, '(A9)') AS 0,9,5//CR
182  C   WRITE(4, '(A9)') AS 0,9,5//CR
183
184      READ(3, '(A512)', IERR=42) MSG
185
186  C -----
187  C   THIS PART LOOKS FOR THE MESSAGE SENT OUT THAT IS ECHOED BACK TO
188  C   THE SERIAL PORT.  ONCE THIS IS READ THE REST OF THE INPUT STRING
189  C   IS DATA.
190  C -----
191      ISTART = 1
192      ICNT = 10                ! STROBE FIRST 9 CHANNELS
193
194 45  DO 50 J = 1, 80          ! SEARCH FOR PART OF THE STRING
195
196      IF (MSG(J:J + 1).EQ.'AS'.OR.MSG(J:J + 1).EQ.'AK') GOTO 70
197
198 50  CONTINUE
199
200  C -----
201  C   THIS IS AN ERROR CATCHING PORTION.  IF NO STRING IS FOUND THEN
202  C   I TRY UP TO 5 TIMES.  IF I STILL GET NOWHERE I SET REPORT = 1
203  C   AND RETURN TO THE CALLER.
204  C -----
205
206      IERR = IERR + 1
207
208      IF (IERR.EQ.6) THEN
209          REPORT = 1
210          GOTO 1000
211      ENDIF
212
213      GOTO 42                ! STROBE THE A/D AGAIN

```

# Appendix IV

```

214
215 C _____
216
217 70 DO 80 I - ISTART,ICNT      ! READ IN THE HEX DATA
218
219     READ(3, '(A6)') A2D_INPUT(I)
220 C   WRITE(4, 'X', ' ', A2D_INPUT(I)) / CR
221
222
223 80 CONTINUE
224
225 C _____
226 C   THIS GOES AND SCANS CHANNELS 7 - 10
227 C _____
228
229     IF (IRESCAN.EQ.0) THEN
230         IRESCAN - 1
231         ISTART - 11
232         ICNT - 11
233
234 85     WRITE(3, '(A8)') AK 10,5 // CR
235 C     WRITE(4, '(A7)') AK 10,5'
236
237     CALL TWAIT(2)
238
239     READ(3, '(A255)', ERR=85) MESC
240
241     GOTO 45
242 ENDIF
243
244 C _____
245 C   THIS PART ASSUMES THAT THE REST OF THE READ STRING IS HEX DATA
246 C   SEPARATED BY CARRIAGE RETURNS AND LINE FEEDS. I USE THESE AS
247 C   BREAKS BETWEEN THE NUMBERS AND PLACE THE INDIVIDUAL NUMBERS
248 C   IN AN ARRAY HEX.
249 C _____
250
251 90 L - 0
252 K - 0
253
254 DO 110 M - 1,11
255
256 C _____
257 C   THIS DETERMINES IF THE DATA IS INDEED HEX OR JUST GARBAGE.
258 C   IF ITS GARBAGE I GO READ AGAIN.
259 C _____
260
261 DO 100 I - 1,6
262
263     IF (A2D_INPUT(M)(I-1).LT.CHAR(48).OR.
264     A2D_INPUT(M)(I-1).GT.CHAR(70)) GOTO 100
265
266     L - L + 1          ! INCREMENT CHARACTER CNT
267

```

# Appendix IV

```

268      CDAT(L:L) - A2D_INPUT(M)(E)      ! PLACE DATA IN ARRAY
269
270      IF(L.EQ.4)THEN                    ! CAN ONLY HAVE 4 CHARA'S
271          IFLG - 0                      ! INITIALIZE DATA FLAG
272          L - 0                        ! RESET CHARACTER COUNTER
273          HEX(M) - CDAT(1:4)           ! PLACE DATA IN ARRAY
274          CDAT - ''
275          GOTO 110
276      ENDIF
277
278 100  CONTINUE
279
280 110  CONTINUE                          ! CONTINUE THRU LOOP
281
282  C-----
283  C   THIS REQUESTS THAT THE HEX CHARACTER DATA BE CONVERTED TO REAL
284  C   DATA TO BE SENT BACK TO THE CALLER.
285  C-----
286
287 135  CALL HEXDEC(HEX,RDAT,ICNT)
288
289      WRITE(3, '(A4)GS//DO '          ! CLOSE EXISTING COMMUNICATIONS
290
291      CALL TWAIT(2)                   ! PAUSE .2 SECS
292
293      WRITE(3, '(A4)GS//DS//CR '      ! CONTROL IS TO SAMPLE STAGE
294
295      CALL TWAIT(2)                   ! PAUSE .2 SECONDS
296
297 1000 RETURN
298      END

```

## Appendix IV

### *AIV.4 Analog APSD Software Modules: DECODE Source Code.*

```

1      SUBROUTINE DECODE (CIN,INTOUT)
2      C-----
3      C   THIS IS DECODE.FOR CALLED BY THE PANEL SUBROUTINE PA.FOR
4      C   IT IS GIVEN THREE BYTES GENERATED BY THE MOUSE THAT WILL
5      C   BE DECODED TO RETURN A PICK ID FOR A PARTICULAR PANEL.
6      C   CIN IS THE 3 CHARACTER INPUT STRING AND INTOUT IS AN
7      C   INTEGER SENT BACK TO THE CALLER.
8      C-----
9
10     CHARACTER *3 CIN
11     C   CHARACTER *(*)CIN
12     INTEGER*4 HI1,HI2,LO1
13     C AN INTEGER REPORT IS ALWAYS ENCODED AS 3 BYTES.
14     I=0
15     I=I+1
16     HI1=ICHAR(CIN(I:1))-32
17     I=I+1
18     HI2=ICHAR(CIN(I:1))-32
19     I=I+1
20     LO1=ICHAR(CIN(I:1))-32
21     I=I+1
22     C   DECODE THE 3 BYTES TO OBTAIN THE INTEGER VALUE.
23     INTOUT=HI1*1024+HI2*16+JMOD(LO1,16)
24     IF(MOD(LO1/16,2).EQ.0) INTOUT=-INTOUT
25     RETURN
26     END

```

## Appendix IV

### AIV.5 Analog APSD Software Modules: DRAW\_ELE Source Code.

```

1      SUBROUTINE DRAW_ELE(SWEEP,ANGLE,RARM,RINCR,REND,RDAT,
2      .INUSE)
3      C-----
4      C   THIS MOD DRAWS A GRAPH OF THE USER SELECTED ELEMENTS FOR EACH
5      C   SWEEP OF THE SAMPLE STAGE. THIS IS INTENDED TO BE SOMEWHAT REAL
6      C   TIME SO USER CONTROL OMITTED.
7      C-----
8
9
10     CHARACTER TEXT*35,TEXT1*35,TEXT2*35,TEXT3*35,TEXT4*35
11     CHARACTER CNUM*10,COL*3,LAST_PT*5,START*6,END*6
12     CHARACTER E,A*5,SEG*3,SPACE*40,CT*6,B,A1*5
13     CHARACTER CURRENT*6,INCREMENT*6,POINT*5
14
15     INTEGER RECENT,REC,IFLG,X,Y,BUSY,MATRIX
16     INTEGER X2,Y2,X3,Y3
17
18     REAL RARM,RINCR,REND,RDAT,RBUF,ANGLE,R1,R2
19
20     DIMENSION LDAT1(10),LDAT2(10),LDAT3(10),LDAT4(10)
21
22     DIMENSION LAST_PT(9),POINT(9),ICHAN(9)
23
24     DIMENSION IA(9),IB(9),IC(9),ID(9),RDAT(16),IELE(16),RBUF(9)
25
26     COMMON LDAT1,LDAT2,LDAT3,LDAT4
27
28     C-----
29     C   THESE ARE THE MATRIX ELEMENT ARRAY NUMBERS THAT ARE COLLECTED
30     C   BY EACH PERMUTATION OF THE POLARIZERS.
31     C-----
32
33     DATA IA / 1,2,4,5,6,8,13,14,16/      ! VERTICAL, VERTICAL
34     DATA IB / 1,2,4,9,10,12,13,14,16/     ! VERTICAL, 45 DEG
35     DATA IC / 1,3,4,5,7,8,13,15,16/      ! 45 DEG , VERTICAL
36     DATA ID / 1,3,4,9,11,12,13,15,16/    ! 45 DEG , 45 DEG
37
38     J   FORMAT (40(' '))
39     WRITE(SPACE,3)
40     E = CHAR(27)
41
42     C-----
43
44     WRITE(*,"J5/LV0"      ! DISABLE DIALOG AREA
45     WRITE(*,"J5/LZ"      ! CLEAR THE DIALOG AREA
46
47     C-----
48     C   THIS SETS UP THE DIALOG AREA SO THE USER HAS AN IDEA WHAT IS BEING
49     C   DISPLAYED. THIS REPORTS THE CURRENT POLARIZER POSITION, THE
50     C   START, END AND INCREMENT OF THE SAMPLE STAGE ANGLE. ADDITIONALLY,
51     C   THE CURRENT ANGLE OF INCIDENCE IS DISPLAYED.

```

# Appendix IV

```

52  C-----
53
54  IF(INUSE.EQ.2)THEN          ! INITIALIZE COUNTERS
55      INUSE = 0
56      INUSE1 = 0
57  ENDIF
58
59  IF (INUSE.EQ.0)THEN          ! ONLY NEED TO ESTABLISH
60
61      CALL CMIX(RARM,START)      ! TEXT ONCE FOR THE WHOLE
62
63      CALL CMIX(REND,END)        ! ROUTINE.
64
65      CALL CMIX(RINCR,INCREMENT) ! START, END AND INCREMENT
66                                  ! ARE MADE CHARACTERS HERE
67      TEXT1 = 'START ANGLE: '//START// END ANGLE: '//END
68      TEXT2 = 'ANGLE INCREMENT: '//INCREMENT
69      TEXT3 = 'CURRENT ANGLE:      Degrees'
70
71  C-----
72  C THIS SAVES THE ELEMENTS THAT WILL BE GRAPHED FOR EACH POLARIZER
73  C PERMUTATION TO AN ARRAY.
74  C-----
75
76  IF(SWEEP.EQ.1)THEN          ! POLARIZERS VERT, VERT
77
78      TEXT = 'POLARIZERS: Vertical, Vertical'
79      DO 70 I=1,LDAT1(10)      ! READ THE A/D CHANNELS
80          ICHAN(I) = LDAT1(I)  ! INTO A BUFFER ARRAY
81  70  CONTINUE
82
83      IEND = LDAT1(10)         ! SAVE COUNT ON ELEMENTS
84
85      DO 75 I = 1,9            ! LOOP THRU MATRIX ELE
86          IELE(I) = IA(I)      ! SAVE IN ARRAY
87  75  CONTINUE
88
89  ELSEIF(SWEEP.EQ.2)THEN
90
91      TEXT = 'POLARIZERS: Vertical, 45 Degrees'
92      DO 80 I=1,LDAT2(10)      ! READ THE A/D CHANNELS
93          ICHAN(I) = LDAT2(I)  ! INTO A BUFFER ARRAY
94  80  CONTINUE
95
96      IEND = LDAT2(10)         ! SAVE COUNT ON ELEMENTS
97
98      DO 85 I = 1,9            ! LOOP THRU MATRIX ELE
99          IELE(I) = IB(I)      ! SAVE IN ARRAY
100  85  CONTINUE
101
102      X = X + INCR
103
104  ELSEIF(SWEEP.EQ.3)THEN
105

```

# Appendix IV

```

106      TEXT - 'POLARIZERS: 45 Degrees, Vertical'
107      DO 90 I=1,LDAT3(10)      ! READ THE A/D CHANNELS
108      ICHAN(I) = LDAT3(I)      ! INTO A BUFFER ARRAY
109  90    CONTINUE
110      IEND = LDAT3(10)      ! SAVE COUNT ON ELEMENTS
111
112      DO 95 I = 1,9      ! LOOP THRU MATRIX ELE
113      IELE(I) = IC(I)      ! SAVE IN ARRAY
114  95    CONTINUE
115      X = X + INCR
116
117      ELSEIF(ISWEEP.EQ.4)THEN
118
119      TEXT - 'POLARIZERS: 45 Degrees, 45 Degrees'
120      DO 100 I=1,LDAT4(10)      ! READ THE A/D CHANNELS
121      ICHAN(I) = LDAT4(I)      ! INTO A BUFFER ARRAY
122  100    CONTINUE
123      IEND = LDAT4(10)      ! SAVE COUNT ON ELEMENTS
124
125      DO 105 I = 1,9      ! LOOP THRU MATRIX ELE
126      IELE(I) = ID(I)      ! SAVE IN ARRAY
127  105    CONTINUE
128      X = X + INCR
129
130      ENDIF
131
132  C-----
133  C  FLAG THE SEE_ELE ROUTINE TO DISPLAY THE ELEMENTS THAT ARE
134  C  POSSIBLE FOR THIS SWEEP
135  C-----
136
137      MATRIX = 1
138      CALL SEE_ELE(ISWEEP,MATRIX,INU)
139
140  C-----
141  C  THIS PART DRAWS THE TEXT TO THE DIALOG AREA WINDOW.
142  C-----
143
144      ISEC = 800
145      CALL INTRPT(ISEC,SEC)
146      WRITE(*,"E//SK//SEC")
147      WRITE(*,"E//SE//SEC")
148      WRITE(*,"E//MT")
149      X2 = 120
150      Y2 = 350
151      CALL HTY(X2,Y2,A1)
152      WRITE(*,"E//LP//A1")
153      WRITE(*,"E//LTB//TEXT")
154
155      Y2 = Y2 - 100
156      CALL HTY(X2,Y2,A1)
157      WRITE(*,"E//MT1")
158      WRITE(*,"E//LP//A1")
159      WRITE(*,"E//LTB//TEXT1")

```

# Appendix IV

```

160
161      Y2 - Y2 - 100
162      CALL HTY(X2,Y2,A1)
163      WRITE(*,"E/LP"//A1)
164      WRITE(*,"E/LTB"//TEXT2)
165
166      Y2 - Y2 - 100
167      CALL HTY(X2,Y2,A1)
168      WRITE(*,"E/LP"//A1)
169      WRITE(*,"E/LTB"//TEXT3)
170      WRITE(*,"E/SC"
171
172
173      ISEG - 850
174      CALL INTRPT(ISEG,SEG)
175      WRITE(*,"E/SK"//SEG)
176      WRITE(*,"E/TKN"
177      CALL TWAIT(10)
178      WRITE(*,"E/SE"//SEG)
179
180      ENDIF
181
182      C-----
183      C   THIS WRITES THE CURRENT INCIDENT ANGLE TO THE DIALOG AREA
184      C-----
185
186      CALL CMX(ANGLE,CURRENT)
187
188      X2 - 800
189      Y2 - 25
190
191      WRITE(*,"E/MP"
192
193      CALL HTY(X2,Y2,A1)
194      WRITE(*,"E/LP"//A1)
195
196      X2 - X2 + 300
197      CALL HTY(X2,Y2,A1)
198      WRITE(*,"E/LG"//A1)
199
200      Y2 - Y2 + 105
201      CALL HTY(X2,Y2,A1)
202      WRITE(*,"E/LG"//A1)
203
204
205      X2 - X2 - 300
206      CALL HTY(X2,Y2,A1)
207      WRITE(*,"E/LG"//A1)
208
209      Y2 - Y2 - 105
210      CALL HTY(X2,Y2,A1)
211      WRITE(*,"E/LG"//A1)
212      WRITE(*,"E/LE"
213

```

# Appendix IV

```

214      Y2 = Y2 + 25
215      CALL HTY(X2,Y2,A1)
216
217      WRITE(*,"E//LP//A1
218      WRITE(*,"E//MT/"
219      WRITE(*,"E//LT6//CURRENT
220      WRITE(*,"E//MT/"
221
222      C-----
223      C   THIS CHECKS TO SEE IF THE USER WANTED TO OMIT DRAWING THE GRAPH
224      C   FOR THIS SPECIFIC SWEEP. THE 10TH ELEMENT CONTAINS THE NUMBER
225      C   OF THE 9 ELEMENTS THAT WILL BE DRAWN.
226      C-----
227
228
229      IF(ISWEEP.EQ.1)THEN
230          IF(LDAT1(10).EQ.0)GOTO 1000      ! NO ELEMENTS ARE DRAWN
231
232      ELSEIF(ISWEEP.EQ.2)THEN
233          IF(LDAT2(10).EQ.0)GOTO 1000      ! NO ELEMENTS ARE DRAWN
234
235      ELSEIF(ISWEEP.EQ.3)THEN
236          IF(LDAT3(10).EQ.0)GOTO 1000      ! NO ELEMENTS ARE DRAWN
237
238      ELSEIF(ISWEEP.EQ.4)THEN
239          IF(LDAT4(10).EQ.0)GOTO 1000      ! NO ELEMENTS ARE DRAWN
240
241      ENDIF
242
243      C-----
244      C-----
245      C   THIS MOD DRAWS THE LINE SEGMENTS TO THE SCREEN
246      C-----
247
248
249
250      DO 200 K = 1, IEND                ! LOOP THRU THE ELEMENTS
251
252      C-----
253      C   THIS PART DETERMINES THE COLOR EACH MATRIX LINE ELEMENT WILL BE
254      C-----
255
256      M = ICHAN(K)                      ! M - THE CHANNEL READ
257      ICOL = IELE(M)                    ! ICOL - THE ELEMENT COLOR
258
259      IF(ICOL.EQ.16)THEN                 ! THIS COLOR IS WHITE
260          WRITE(*,"E//MV1"              ! MAKE LINE DASHED
261          WRITE(*,"E//ML1"              ! LINE COLOR WHITE
262      ELSE
263          WRITE(*,"E//MV0"              ! MAKE LINE SOLID
264          CALL INTRPT(ICOL,COL)          ! INTEGER TO TEK CHARA
265          WRITE(*,"E//ML//COL(1:1)      ! WRITE COLOR TO TERMINAL
266      ENDIF
267      C-----

```

# Appendix IV

```

268 C THE X IS THE BEGINNING ANGLE OF THE SAMPLE FOR WHICH DATA IS
269 C RETRIEVED. THE ZERO POINT IS SCREEN POSITION 245. EACH DEGREE
270 C INCREMENT IS 20 SCREEN UNITS. THUS THE EQUATION BELOW:
271 C _____
272
273 IF(INUSE1.EQ.0)THEN ! USE ONLY ONCE ON RUN
274
275
276 X = IPIX(RARM)
277 X = 245 + (20 * X)
278
279 C _____
280 C THE INCREMENT FOR EACH DATA PT IS IN INCR. HERE NORMALIZED TO
281 C SCREEN UNITS. 1 DEG = 20 SCREEN UNITS.
282 C _____
283
284 INCR = IPIX(RINCR)
285 INCR = INCR * 20
286 INUSE1 = 1
287 ENDIF
288
289 C _____
290
291 K1 = 100
292 IYPOINT = 0
293
294 C _____
295 C THIS MOD RESOLVES THE VOLTAGE INTO A Y COORDINATE ( IYPOINT )
296 C THE BEST RESOLUTION IS 1/1000 VOLT. 1/10 VOLT = 100 SCREEN UNITS
297 C 1/100 VOLT = 10 SCREEN UNITS AND 1/1000 VOLT = 1 SCREEN UNIT.
298 C _____
299
300 R2 = RDATA(M) ! SAVE DATA IN BUFFER
301
302 DO 180 N = 1,3
303 R1 = R2 * 10.0 ! READ PROPER CHANNEL
304 INUM = IPIX(R1)
305 IYPOINT = INUM * K1 + IYPOINT
306 R2 = R1 - (FLOAT(INUM))
307 IF(N.EQ.1)K1 = K1 - 90
308 IF(N.EQ.2)K1 = K1 - 9
309 180 CONTINUE
310
311 C _____
312 C THIS MOD NORMALIZES THE SCREEN UNIT WITH THE BAR GRAPH AND DRAWS
313 C THE LINE FROM ONE POINT TO THE NEXT. 1900 IS THE Y COORDINATE FOR
314 C THE 0.0 LINE ON THE Y AXIS.
315 C _____
316
317 Y = 1900 + IYPOINT ! CALCULATE THE Y POINT
318
319 IF(Y.GE.2900)Y = 2900 ! DON'T LET Y LEAVE THE
320 IF(Y.LE.900)Y = 900 ! GRAPH
321 CALL HTY (X,Y,A) ! CALC X,Y VECTOR

```

# Appendix IV

```

322     LAST_PT(M) = A           ! SAVE IN BUFFER ARRAY
323
324     IF(INUSE.EQ.0)THEN        ! IF THIS IS THE 1ST POINT
325         WRITE(*,"E//ML2"
326         WRITE(*,"E//MM2"      ! SET MARKER MODE TO "+"
327         WRITE(*,"E//LP//A"    ! SET THE ORIGIN
328         WRITE(*,"E//LH//A"    ! DRAW THE MARKER
329     ELSE
330
331     C _____
332     C THIS PART MUST ESTABLISH AN ORIGIN THAT WAS THE LAST POINT BEFORE
333     C EVERY DRAW.
334     C _____
335
336
337     WRITE(*,"E//LP//POINT(M)" ! SET THE ORIGIN
338     WRITE(*,"E//LG//A"        ! DRAW TO NEW POINT
339
340     ENDIF
341
342 200 CONTINUE
343
344     C _____
345     C THIS INCREMENTS THE COUNTER FOR THE NEXT X
346     C _____
347
348     IF(ISWEEP.EQ.1.OR.ISWEEP.EQ.3)THEN
349         X = X + INCR          ! INCREMENT THE X POINT
350     ELSE
351         X = X - INCR          ! DECREMENT THE X VALUE
352     ENDIF
353
354     C _____
355     C THIS SAVES THE VECTORS OF EACH LINE ELEMENT CALCULATED FOR THIS
356     C POINT INTO A BUFFER SO THAT THEY CAN BE USED AS ORIGINS FOR THE
357     C NEXT GROUP OF POINTS
358     C _____
359
360     DO 300 I = 1,9
361         POINT(I) = LAST_PT(I)
362 300 CONTINUE
363
364     WRITE(*,"E//MV0"          ! MAKE LINE SOLID
365
366     INUSE = 1                  ! SET INUSE FLAG TO ON
367
368 1000 RETURN
369     END

```

# Appendix IV

## AIV.6 Analog APSD Software Modules: GET\_ARRAY Source Code.

```

1  SUBROUTINE GET_ARRAY(LET,REDRAW)
2
3  C-----
4  C  THIS ROUTINE IS USED TO GET THE USER TO SELECT AN ARRAY
5  C  ELEMENT.
6  C  LET - THE CHARACTER LETTER A - TO BE SELECTED BY THE USER
7  C  REDRAW - IS A FLAG FOR THIS ROUTINE TO REDRAW THE INVISIBLE ARRAY
8  C  BLOCKS. WHEN SET TO - 1
9  C-----
10
11  CHARACTER E*1,LET*1,SEG*3,ANS*12
12
13  INTEGER X,Y,REDRAW
14
15  E = CHAR(27)
16
17
18  IF(REDRAW.EQ.1)THEN      ! FLAG TO TURN ON ALL 5 BLOCKS
19  REDRAW = 0              ! RESET THE FLAG
20
21  IF(KCHAR(LET).GT.64.OR.KCHAR(LET).LT.70)THEN
22  J = KCHAR(LET) - 33      ! SEGMENT NUMBER TO START WITH
23
24  DO 10 I = 32,36          ! ONE BLOCK IS ALREADY ON
25  IF(I.EQ.J)GOTO 10        ! IF ON SCREEN DON'T DRAW
26  CALL INTRPT(I,SEG)      ! CONVERT TO TEK CHARACTER
27  WRITE(*,"E//SV//SEG(1:2)/I" ! DRAW THE BOX
28  10  CONTINUE
29  ENDIF
30  ENDP
31
32  X = 1500                  ! PLACE THE MOUSE X,Y
33  Y = 350
34  IFLAG = 1
35
36  CALL GIN(X,Y,IFLAG,IMODE,ITYPE IGIN,IPOINT)
37
38  15  WRITE(*,"E//LV0"      ! SET UP THE DIALOG AREA
39  WRITE(*,"E//L144"        ! COLOR WHITE ON BLUE
40  WRITE(*,"E//L2"         ! CLEAR OF TEXT
41  WRITE(*,"E//LV1"        ! ENABLE DIALOG AREA
42
43
44  WRITE(*,"Y SELECT:"
45  WRITE(*,"Y 1. AN ARRAY ELEMENT"
46  WRITE(*,"Y
47  WRITE(*,"Y 3. EXIT"
48
49  20  READ(*,"(A12)",ERR=15)ANS      ! READ FROM THE MOUSE
50
51  IF(ANS(1:1) EQ '1'.OR.ANS(1:1).EQ '3')GOTO 30

```

# Appendix IV

```

52      GOTO 20
53
54      C _____
55      C   THIS CHECKS THE SELECTION TO MAKE SURE AN ARRAY ELEMENT WAS
56      C   CHOSEN.
57      C _____
58
59      30  SEG = ANS(7:9)      ! GET USER CHARACTER SELECTION
60      CALL DECODE(SEG,ISEG)  ! CHANGE SEGMENT NUMBER TO INTEGER
61
62      IF(ISEG + 33.GT.69.OR.ISEG + 33.LT.65)GOTO 20
63
64      WRITE(*,"E/ID"        ! DELETE THE GIN CURSOR
65      WRITE(*,"E//SKC)")
66
67
68      LET = CHAR(ISEG + 33)   ! PLACE THE SELECTED LETTER IN LET
69
70      C _____
71      C   THIS PART ERASES THE REST OF THE BLOCKS THAT ARE NOT IN USE
72      C   ISEG IS THE SELECTED SEGMENT NUMBER.
73      C _____
74
75
76      DO 40 I = 32,36        ! LOOP THRU THE SEGMENT NUMBERS
77      IF(I.EQ.ISEG)GOTO 40   ! IF ON SCREEN DON'T DRAW
78      CALL INTRPT(I,SEG)     ! CONVERT TO TEK CHARACTER
79      WRITE(*,"E//SV//SEG(1:2)/r") ! ERASE THE BOX
80      40  CONTINUE
81
82      RETURN
83      END

```

## Appendix IV

### AIV.7 Analog APSD Software Modules: GIN Source Code.

```
1      SUBROUTINE GIN(X1,Y1,IFLAG,IMODE,ITYPE,IGIN,IPOINT)
2
3      C          CHARLES HENRY JULY 7 1989
4      C-----
5      C  THIS PROGRAM IS CALLED GIN.FOR. IT WILL DEFINE A "GIN"( GRAPHICS
6      C  INPUT, OUTPUT ) CURSORFOR USE IN PROGRAM.
7      C  THIS WILL EITHER ENABLE THE MOUSE,PUCK OR THUMBWHEELS.
8      C  THE DEFAULT IS SET FOR THE MOUSE USING THE ARROW CURSOR IN PICK
9      C  MODE
10     C
11     C  X1,Y1 - NEW LOCATION FOR THE CURSOR.  DEFINED BY USER
12     C
13     C  IFLAG - FLAG THAT MAKES ALL SEGMENTS VISIBLE. THIS IS USEFUL WHEN
14     C          THE USER IS BUILDING SOMETHING WITH ALL SEGMENTS TURNED
15     C          OFF AND THE GIN SUBROUTINE IS CALLED LAST.  THE DEFAULT
16     C          IS TO ISSUE THE COMMAND EVEN IF NOTHING IS OFF
17     C
18     C  IMODE - THIS IS THE GIN MODE.  THERE ARE THREE TYPES
19     C          EACH TYPE LETS YOU PERFORM SLIGHTLY DIFFERENT TASKS.
20     C
21     C          PICK RETURNS BUTTON PRESSED,X/Y LOCATION,VIEW NUMBER AND
22     C          WHICH SEGMENT WAS PICKED.
23     C
24     C          LOCATE RETURNS A SINGLE REPORT CONTAINING THE BUTTON
25     C          PRESSED,X/Y LOCATION,AND VIEW NUMBER IF SPECIFIED.
26     C          SINCE THE LOCATE MODE FOR THIS PROGRAMS USE IS NOT MUCH
27     C          DIFFERENT THEN THE PICK.  I AM USING IT HERE TO DRAW AND
28     C          USE GIN RUBBERBANDING.
29     C
30     C          STROKE RETURNS ONE OR MORE FIRST POINTS WHICH ARE SEPARATED
31     C          BY DIFFERENT FIRST CHARACTERS,AND A LAST POINT.THE FIRST
32     C          REPORT WILL INCLUDE THE BUTTON SELECTED THEN THE SUBSEQUENT
33     C          STROKE POINT CHARACTERS WILL BE A " J " FOLLOWED BY THE X/Y
34     C
35     C          THE LAST POINT WILL BE A " O " FOLLOWED BY THE X/Y.
36     C
37     C          0 - PICK
38     C          1 - LOCATE
39     C          2 - STROKE
40     C
41     C
42     C  ITYPE - THIS IS THE CURSOR SEGMENT TYPE.  NORMALLY ITS AN ARROW
43     C          BUT THERE ARE OCCASSIONS WHEN A CROSSHAIR IS USEFUL.
44     C          ADDITIONALLY, WHEN THE GIN IS ACTIVE AND THE USER IS
45     C          SELECTING A MENU ITEM THAT REQUIRES NO GIN DEVICE AN
46     C          ALTERNATE SEGMENT IS PROVIDED THAT IS A DOT.  IF THE
47     C          USER ALREADY HAS A SEGMENT THAT IS NEEDED FOR THE CURSOR
48     C          THEN A J IS PASSED IN ITYPE AND THE INTEGER VALUE IS
49     C          PASSED IN IPOINT.
50     C
51     C          0 - ARROW SEGMENT
```

## Appendix IV

```

52 C      1 - CROSSHAIR
53 C      2 - DOT
54 C      3 - NUMBER OF SEGMENT TO BE MADE THE CURSOR
55 C
56 C      IGIN - THIS IS THE GRAPHICS INPUT DEVICE. THERE CAN BE FOUR OR
57 C      FIVE DIFFERENT DEVICES. THE ONES WE USE ARE LISTED BELOW
58 C
59 C      0 - MOUSE
60 C      1 - THUMBWHEELS
61 C      2 - PUCK (LARGE TABLET) ..(THIS IS SPECIFIED BY THE PORT)
62 C
63 C-----
64
65      INTEGER X,Y,X1,Y1
66      CHARACTER*5 A,A1,A2,A3,A4,A5,A6
67      CHARACTER E,AAA*12,C*2,GS,US,SEG*3,A10*6,B*1
68
69      E=CHAR(27)
70      GS=CHAR(29) ! START VECTOR MODE CHAR
71      US=CHAR(31) ! STOP VECTOR MODE CHAR
72
73 C-----
74 C      THE FOLLOWING IS THE SETUP FOR THE PARTICULAR GIN DEVICE THE
75 C      USER IS REQUESTING. exp...USER MAY WANT THE MOUSE TO MOVE THE
76 C      CROSSHAIR IN STROKE MODE.
77 C
78 C
79 C      IGIN - 0      MOUSE
80 C      ITYPE - 1     CROSSHAIR
81 C      IMODE - 2     STROKE MODE
82 C
83 C      BELOW WE START DEFINING WITH THE PICK FUNCTION
84 C      I HAVE TO DEFINE THE NUMBER OF CHARACTERS IN THE G STRING"" SO
85 C      THAT THERE WONT BE ANY INNER SPACING IN THE ESCAPE
86 C      COMMANDS. THAT WOULD CAUSE A FATAL ERROR. THIS IS STORED IN "12"
87 C
88 C-----
89
90      WRITE(*,*)E//ID*      ! DELETE THE GIN SEGMENT
91      WRITE(*,*)E//SKC1*    ! DELETE THE GIN CURSOR
92      WRITE(*,*)E//LV*      ! DISABLE DIALOG AREA
93
94      IF(IMODE.EQ.0)THEN      ! USER WANTS PICK FUNCTION
95      IF(IGIN.EQ.0)THEN      ! SET UP FOR MOUSE
96      C = 'D1'
97      I2 = 2      ! NUMBER CHARA USED IN STRING
98      ELSEIF(IGIN.EQ.1)THEN
99      C = '1'      ! SET FOR THUMBWHEELS
100     I2 = 1      ! NUMBER CHARA USED IN STRING
101     ELSEIF(IGIN.EQ.2)THEN
102     IF(PORT.EQ.0)C = 'A9' ! TABLET PORT 0
103     IF(PORT.EQ.1)C = 'B1' ! TABLET PORT 1
104     IF(PORT.EQ.2)C = 'B9' ! TABLET PORT 2
105     I2 = 2      ! NUMBER CHARA USED IN STRING

```

# Appendix IV

```

106      ENDP
107
108  C-----
109  C   HERE WE BEGIN WITH THE LOCATE FUNCTION
110  C-----
111
112      ELSEIF(IMODE.EQ.1)THEN      ! USER WANTS THE LOCATE FUNCTION
113      IF(ICIN.EQ.0)THEN          ! SET UP FOR MOUSE
114          G - 'D0'
115          IZ - 2                ! NUMBER CHARA USED IN STRING
116      ELSEIF(ICIN.EQ.1)THEN
117          G - '0'               ! SET FOR THUMBWHEELS
118          IZ - 1                ! NUMBER CHARA USED IN STRING
119      ELSEIF(ICIN.EQ.2)THEN
120          IF(IPORT.EQ.0)G - 'A0' ! TABLET PORT 0
121          IF(IPORT.EQ.1)G - 'B0' ! TABLET PORT 1
122          IF(IPORT.EQ.2)G - 'B0' ! TABLET PORT 2
123          IZ - 2                ! NUMBER CHARA USED IN STRING
124      ENDP
125
126  C-----
127  C   THIS PART DEFINES THE STROKE FUNCTIONS
128  C   THESE ARE ONLY ALLOWED FOR THE TABLET AND THE MOUSE.
129  C
130  C-----
131
132
133      ELSEIF(IMODE.EQ.2)THEN      ! USER WANTS THE STROKE FUNCTION
134      IF(ICIN.EQ.0)THEN          ! SET UP FOR MOUSE
135          G - 'D2'
136          IZ - 2                ! NUMBER CHARA USED IN STRING
137      ELSEIF(ICIN.EQ.1)THEN
138          G - ':'               ! SET FOR THUMBWHEELS
139          IZ - 1                ! NUMBER CHARA USED IN STRING
140      ELSEIF(ICIN.EQ.2)THEN
141          WRITE(*,*)'YOU MUST NOT SPECIFY A PORT FOR STROKE'
142          GOTO 1000
143      ENDP
144      ENDP
145
146  C-----
147
148
149      WRITE(*,*)'E/SV=0'         ! MAKE CURSOR INVISIBLE
150
151  4  IF(ITYPE.EQ.1.OR.ITYPE.EQ.3)GOTO 30 ! DON'T DRAW A SEGMENT
152
153  C-----
154  C   THE USER IS REQUESTING THAT A DOT BE DRAWN INSTEAD OF THE ARROW
155  C-----
156
157      IF(ITYPE.EQ.2)THEN          ! THIS IS THE DOT SEGMENT
158          WRITE(*,*)'E/MLO/E/MTO/E/MMO'
159

```

## Appendix IV

```

160 C-----
161 C THE SEGMENT IS A SMALL INVISIBLE MARK THAT IS USED WHEN THE USER
162 C IS SELECTING MENU ITEMS FROM THE DIALOG AREA. ALL THAT IS
163 C SEARCHED FOR IS THE BUTTON THAT WAS PRESSED.
164 C-----
165
166
167 WRITE("E//SOC11" ! BEGIN SEGMENT NO. 4049
168 CALL HTY(X1,Y1,A) ! PLACE THE SEGMENT AT X,Y LOCAL
169 WRITE("E//LH//A" ! DRAW THE SEGMENT
170 WRITE("E//SC" ! CLOSE THE SEGMENT
171 GOTO 30 ! JUMP OVER ARROW DRAW
172 ENDIF
173
174 C-----
175 C THIS BEGINS THE DRAWING OF THE ARROW CURSOR.
176 C-----
177
178 S CONTINUE
179 WRITE("E//SC"
180 X=0
181 Y=100
182 CALL HTY(X,Y,A)
183 WRITE("E//SP//A" ! SET PIVOT POINT FOR CURSOR
184
185 WRITE("E//SOC11"
186 WRITE("E//MP//E//ML1//E//MT1"
187
188 C-----
189 C THIS PART DRAWS THE ARROW SEGMENT. IT IS DRAWN AT THE 0,0 ORIGIN
190 C AND THEN IS MOVED TO THE USER DEFINED LOCATION.
191 C-----
192
193 X=0
194 Y=100
195 CALL HTY(X,Y,A)
196 X=80
197 Y=70
198 CALL HTY(X,Y,A1)
199 X=40
200 Y=70
201 CALL HTY(X,Y,A2)
202 X=102
203 Y=12
204 CALL HTY(X,Y,A3)
205 X=95
206 Y=0
207 CALL HTY(X,Y,A4)
208 X=30
209 Y=60
210 CALL HTY(X,Y,A5)
211 X=30
212 Y=20
213 CALL HTY(X,Y,A6)

```

# Appendix IV

```

214
215 WRITEC,"E/FP//A/P//GS//A1//A2//A3//A4//A5//A6//
216 US/E/SC"
217
218 20 CONTINUE
219
220 30 IF(ITYPE.EQ.1)GOTO 35
221 WRITEC,"E/SC10" ! MAKE CURSOR INVISIBLE
222 35 WRITEC,"E/SC1" ! MAKE CURSOR VISIBLE
223
224 IF(MODE.EQ.1)THEN ! USER WANTS THE LOCATOR MODE
225
226 IF(ITYPE.EQ.0.OR.ITYPE.EQ.2)THEN
227
228 WRITEC,"E/FP//G(1:2)/P1"//
229 E/FP//G(1:2)/P1//E/FP//G(1:2)/P1"//
230 E/FP//G(1:2)/P0
231 ELSE
232 WRITEC,"E/FP//G(1:2)/P1"//
233 E/FP//G(1:2)/P1//E/FP//G(1:2)/P0
234 ENDP
235
236
237 ELSEIF(MODE.EQ.2)THEN ! USER WANTS STROKE MODE
238
239 IF(ITYPE.EQ.0.OR.ITYPE.EQ.2)THEN
240 WRITEC,"E/FP//G(1:2)/P1"//
241 E/FP//G(1:2)/P0P4//E/FP//G(1:2)/P0
242 ELSE
243 WRITEC,"E/FP//G(1:2)/P0"//
244 E/FP//G(1:2)/P0
245 ENDP
246
247 ELSE ! USER WANTS PICK MODE
248
249 IF(ITYPE.EQ.0.OR.ITYPE.EQ.2)THEN
250 WRITEC,"E/FP//G(1:2)/P1"//
251 E/FP//G(1:2)/P0
252
253 ELSEIF(ITYPE.EQ.3)THEN ! USER WANTS A SEGMENT TO BE
254 CALL INTRPT(PORT,SEG) ! THE GIN CURSOR
255 WRITEC,"E/FP//G(1:2)/SEG"//
256 E/FP//G(1:2)/P0//E/FP//G(1:2)/SEG/P1
257
258 ELSE
259
260 WRITEC,"E/FP//G(1:2)/P0" ! SET MOUSE FOR PICK
261 ! FUNCTIONS
262 ENDP
263 ENDP
264
265 C-----
266
267 CALL HRY(X1,Y1,A)

```

# Appendix IV

```

268
269      IF (TYPE.EQ.0) THEN          ! IF ITS SEGMENT
270          WRITE(*,"E/PSXC11//A"    ! POSITION THE CURSOR
271      ELSEIF (TYPE.EQ.1) THEN      ! IF ITS CROSS HAIR
272          WRITE(*,"E/PSX0//A"      ! POSITION THE CURSOR
273      ELSEIF (TYPE.EQ.3) THEN      ! IF ITS CROSS HAIR
274          WRITE(*,"E/PSX//SEG//A"  ! POSITION THE CURSOR
275      ENDIF                        ! THE DOT IS ALREADY
276                                  ! PLACED
277
278 1000  ICUR=0                      ! RESET THE TOGGLE LOOP
279      IMODE = 0                    ! RESET THE GIN TYPE FLAG
280      ITYPE = 0                    ! RESET THE CURSOR TYPE TO ARROW
281
282      IF (TYPE.EQ.1) GOTO 1010
283      WRITE(*,"E/SVC11"           ! MAKE CURSOR INVISIBLE
284 1010  WRITE(*,"E/1LZ"            ! CLEAR DIALOG AREA
285      RETURN
286      END

```

# Appendix IV

## AIV.8 Analog APSD Software Modules: HEXDEC Source Code.

```

1      SUBROUTINE HEXDEC(HEXIN,RDAT,CHANNELS)
2
3      C -----
4      C   THIS PROGRAM WILL RECEIVE INPUT FROM THE A/D CONVERTER
5      C   CONVERT IT TO BINARY, CHECK TO SEE IF NUMBER IS POSITIVE
6      C   OR NEGATIVE. IF POS THE NUMBER WILL BE CONVERTED TO DECIMAL
7      C   AND SAVED. IF NEG, A 1 WILL BE SUBTRACTED AND THE BITS
8      C   REVERSED (2's COMPLIMENT) AND THEN RETURNED TO DECIMAL.
9      C
10     C   HEXIN - ARRAY OF HEX CHARACTERS PASSED IN FROM THE CALLER
11     C
12     C   RDAT - ARRAY OF DECIMAL REALS PASSED OUT TO THE CALLER
13     C
14     C   CHANNELS - THE NUMBER OF ARRAY ELEMENTS PASSED IN FROM THE CALLER
15     C -----
16
17     CHARACTER*4 HEXIN,A,PLACE*1
18
19     INTEGER K,ICNT,BIN,CHANNELS
20
21     REAL RDAT,RT
22
23     DIMENSION BIN(16),PLACE(16),HEXIN(16),IDATA(16),RDAT(16)
24
25     DATA (BIN(I),I=1,16)/32768,16384,8192,4092,2048,
26     . 1024,512,256,128,64,32,16,8,4,2,1/
27
28
29
30     1  FORMAT (A4)
31     2  FORMAT (T10,8(A1,A1))
32
33     C ----- TEST DATA -----
34     C
35     C   I - 1
36     C
37     C3  WRITE(*,*) 'ENTER UP TO 16 HEX NUMBERS: ( 4 CHAR MAX )'
38     C   WRITE(*,*) 'ENTER " Z " WHEN DONE'
39     C   WRITE(*,*)
40     C   READ(*, '(A4)',ERR = 1000)HEXIN(I)
41     C
42     C   IF(HEXIN(I)(1:1).EQ.'Z'.AND.I.EQ.1)GOTO 1000
43     C   IF(HEXIN(I)(1:1).EQ.'Z')GOTO 10
44     C
45     C   CHANNELS - 1
46     C
47     C   I - I + 1
48     C   GOTO 3
49     C
50     C -----
51

```

# Appendix IV

```

52
53 10 DO 200 KK-1,CHANNELS
54
55 ICNT=0
56 J=0
57 K=0
58 IBUFF=0
59 A=HEXIN(KK)
60 J=-1
61
62 DO 40 II=4,1,-1
63 J=J+1
64
65 IF(ICHAR(A(II:II)).GT.64) THEN
66 K=ICHAR(A(II:II))-55
67 GOTO 35
68 ENDIF
69
70 K=ICHAR(A(II:II))-48
71
72 35 ICNT=ICNT+K*16**J
73
74 40 CONTINUE
75
76 IBUFF=ICNT
77
78 C-----
79 C THE FOLLOWING MOD CHANGES THE INTEGERS TO BINARY NUMBERS
80 C ONLY THE 16TH PLACE HAS TO BE LOOKED AT. IF THIS NUMBER IS
81 C '1' THEN WE HAVE A NEGATIVE NUMBER AND WE WILL DO A TOTAL
82 C CONVERSION.
83 C-----
84
85 IF(ICNT.LT.BIN(1)) GOTO 100
86
87 C-----
88 C IF THE NUMBER IS NEG ALL THE 1'S AND 0'S ARE REVERSED
89 C-----
90
91 IPUT=17
92
93 DO 70 II=1,16
94
95 IPUT=IPUT-1
96
97 IF(IBUFF.GE.BIN(II)) THEN
98 IBUFF=IBUFF-BIN(II)
99 PLACE(IPUT)='0'
100 ELSE
101 PLACE(IPUT)='1'
102 ENDIF
103
104 70 CONTINUE
105

```

# Appendix IV

```

106 C _____
107 C THIS WRITES THE BINARY NUMBER TO THE SCREEN
108 C _____
109 C
110 C WRITE(*,2)PLACE(16),PLACE(15),PLACE(14),PLACE(13),PLACE(12),
111 C 1 PLACE(11),PLACE(10),PLACE(9),PLACE(8),PLACE(7),PLACE(6),PLACE(5),
112 C 1 PLACE(4),PLACE(3),PLACE(2),PLACE(1)
113 C _____
114 C THIS PART CONVERTS THE NUMBER BACK TO DECIMAL, CHANGES THE
115 C SIGN AND THEN SUBTRACTS 1
116 C _____
117
118 JJ = 1
119 ICNT = 0
120
121 DO 90 II=16,1,-1
122
123 IF(PLACE(II).EQ.'1') THEN
124 ICNT = ICNT + BIN(JJ)
125 ENDIF
126
127 JJ = JJ + 1
128
129 90 CONTINUE
130
131 ICNT = ICNT - ( 2 * ICNT )
132 ICNT = ICNT - 1
133 100 IDATA(KK) = ICNT
134
135 200 CONTINUE
136
137 C _____
138
139 DO 300 I = 1,CHANNELS
140
141 IF(IDATA(I).EQ.0.0)THEN
142 IDAT(I) = 0.000
143 GOTO 300
144 ENDIF
145
146 IDATA(I) = IDATA(I) * -1
147
148 IF(IDATA(I).GT.0)THEN
149 IDATA(I) = 32676 - IDATA(I)
150 ELSE
151 IDATA(I) = -(IDATA(I) + 32676)
152
153 ENDIF
154
155 C IDAT(I) = (FLOAT(IDATA(I))/16384.0) + 0.0022 ! NORMALIZE VOLTAGE
156
157 IDAT(I) = (FLOAT(IDATA(I))/8192.0) + 0.0022 ! NORMALIZE VOLTAGE
158 C IDAT(I) = (FLOAT(IDATA(I))/4096.0) + 0.0022 ! NORMALIZE VOLTAGE
159

```

# Appendix IV

```

160 300 CONTINUE
161
162
163 C _____
164 C NO NEED TO GO ANY FURTHER IF THE DC ELEMENT - 0
165 C _____
166
167 C IF(RDAT(1).LE.0.0)GOTO 1000 ! ERROR CONDITION
168
169 RT = 1 / RDAT(1) ! NORMALIZATION FACTOR
170
171 C _____
172 C THIS NORMALIZES ALL THE ELEMENTS WITH THE " 11 " ELEMENT
173 C _____
174
175 DO 400 I = 1, CHANNELS
176
177 RDAT(I) = RDAT(I) * RT
178
179 400 CONTINUE
180
181 C _____
182
183 1000 RETURN
184 END

```

## Appendix IV

### AIV.9 Analog APSD Software Modules: HTY Source Code.

```
1      SUBROUTINE HTY(X,Y,A)
2      C-----
3      C  THIS ROUTINE IS A UTILITY THAT CONVERTS TWO INTEGER (X,Y) SCREEN
4      C  COORDINATES INTO TEKTRONICS CHARACTER CODE SCHEME. THE TEK
5      C  TERMINALS HAVE THEIR OWN GRAPHICS LANGUAGE. THIS LANGUAGE REQUIRES
6      C  THE HOST TO ISSUE EVERY COMMAND AS A CHARACTER STRING. IN THIS
7      C  CASE THE TWO VECTORS (X,Y) ARE CONVERTED INTO A FIVE CHARACTER
8      C  STRING. THESE REPRESENT BIT POSITIONS FOR THE TERMINAL AS FOLLOWS:
9      C  HTY, Extra, LoY, HiX, and LoX.
10     C      THIS UTILITY IS ONE OF THE MOST IMPORTANT UTILITIES FOR
11     C  ANY HOST TO TEKTRONICS GRAPHICS APPLICATIONS.
12     C  NOTE: X,Y REMAIN UNCHANGED WHEN SENT BACK TO CALLER.
13     C-----
14
15     INTEGER *2 X,X0,X1,X2,Y,Y0,Y1,Y2
16     CHARACTER *5 A
17
18     X2=X/128
19     Y2=Y/128
20     X1=(X-X2*128)/4
21     Y1=(Y-Y2*128)/4
22     X0=MOD(X,4)
23     Y0=MOD(Y,4)
24
25     C-----
26     C  HERE THE INTEGERS ARE CHANGED TO A CHARACTER STRING.
27     C-----
28
29     A(1:1)=CHAR(32+Y2)
30     A(2:2)=CHAR(96+Y0*4+X0)
31     A(3:3)=CHAR(96+Y1)
32     A(4:4)=CHAR(32+X2)
33     A(5:5)=CHAR(64+X1)
34
35     RETURN
36     END
```

# Appendix IV

## AIV.10 Analog APSD Software Modules: INT\_TO\_CHAR Source Code.

```

1      SUBROUTINE INT_TO_CHAR(NUMBER,CNUM,LENGTH)
2
3      C -----
4      C   THIS ROUTINE WILL CHANGE INTEGERS FROM +/- 99,999,999 INTO
5      C   CHARACTERS.
6      C
7      C   NUMBER - INTEGER PASSED IN FROM THE CALLER
8      C
9      C   CNUM   - CHARACTER REPRESENTATION OF THE ABOVE INTEGER THAT IS
10     C           RETURNED TO THE CALLER.
11     C
12     C   LENGTH - THE NUMBER OF CHARACTERS BEING SENT BACK TO THE CALLER
13     C -----
14
15     CHARACTER *9 CNUM,CBUF
16
17     5      CNUM = ''           ! INITIALIZE CHARACTER VALUE
18
19     C   WRITE(*,*) 'ENTER A NUMBER:'
20     C   WRITE(*,*)
21     C   READ(*,*)NUMBER
22
23     NUMBUF = NUMBER
24     IF(NUMBER.LT.0)THEN        ! ITS A NEGITIVE NUMBER
25         NEG = 1                ! SET A FLAG THAT THE NUMBER IS
26         NUMBER = ABS(NUMBER)    ! NEGITIVE. TAKE ITS ABSOLUTE
27     ENDIF
28
29     IF(NUMBER.EQ.0)THEN        ! THE NUMBER IS ZERO
30         CNUM = '000000000'      ! THE CHARACTER - ALL 0's
31         GOTO 1000              ! RETURN TO CALLER
32
33     ELSEIF(NUMBER.LT.10)THEN    ! ITS A SINGLE DIGIT NUMBER
34         IN = 1                  ! CHARACTER PLACE COUNTER
35         CNUM(1:1) = CHAR(NUMBER + 48) ! CONVERT NUMBER TO CHARACTER
36         GOTO 110
37
38     ELSEIF(NUMBER.GT.999999999.OR.NUMBER.LT.-999999999)THEN
39         CNUM = '000000000'      ! IF THE NUMBER IN IS TO LARGE
40         GOTO 1000              ! CNUM CANT CONVERT THIS NUMBER
41     ENDIF
42
43     C -----
44     C   THIS PART SEPARATES EACH INTEGER PLACE VALUE INTO ITS INDIVIDUAL
45     C   COMPONENTS. IT THEN CONVERTS EACH COMPONENT INTO A CHARACTER
46     C   AND PLACES THAT CHARACTER IN THE CNUM STRING.
47     C -----
48
49     M = 0                      ! PLACE VALUE BUPPER
50     J = 1                      ! PLACE VALUE
51     IN = 0                     ! CHARACTER PLACE COUNTER

```

# Appendix IV

```

52
53      DO 10 I = 1,8          ! LOOP THRU THE PLACE VALUES
54      IF(NUMBER.LT.J)GOTO 20  ! LOOKING FOR THE PLACE VALUE
55      J = J * 10             ! INCREMENT PLACE VALUE
56 10   CONTINUE
57
58 C-----
59 C   HERE I GET THE ACTUAL INTEGER VALUE THAT RESIDES IN EACH PLACE
60 C   VALUE. ie... THOUSANDS PLACE (PLACE VALUE) - 5
61 C               HUNDREDS PLACE (PLACE VALUE) - 3
62 C               TENS   PLACE      - 9
63 C               ONES   PLACE      - 2
64 C-----
65
66 20   I = I - 2
67      J = 10 ** I * 9        ! J GOES TO TOP OF THIS PLACE
68      K = 10 ** I            ! K - BOTTOM OF THIS PLACE VALUE
69
70      DO 100 LOOP = 1, I + 1  ! LOOP THRU THE PLACE VALUES
71
72      DO 40 NUM = J, 0, -K      ! FIND THE INTEGER FOR THIS PLACE
73      L = NUM - NUMBER
74      M = M + 1                ! ACTUAL INTEGER VALUE WANTED
75      IF(L.LE.0)GOTO 30        ! VALUE IS FOUND
76 40   CONTINUE
77
78 C-----
79 C   THIS BUILDS THE CHARACTER STRING FROM EACH PLACE VALUE INTEGER
80 C-----
81
82 50   IN = IN + 1              ! CHARACTER PLACE COUNTER
83      NUM = 10 - M             ! ACTUAL INTEGER SOUGHT
84      CNUM(IN:IN) = CHAR(NUM + 48) ! CONVERT INTEGER TO CHARACTER
85
86      M = 0                    ! RESET THE INTEGER COUNTER
87      NUMBER = NUMBER - (K * NUM) ! DECREMENT THE NUMBER
88      I = I - 1
89      J = 10 ** I * 9          ! DECREMENT TOP VALUE OF PLACE
90      K = 10 ** I              ! DECREMENT BOTTOM VALUE OF PLACE
91
92 100  CONTINUE                 ! END OF LOOP
93
94 C-----
95 C   IF THE NUMBER WAS NEGITIVE I PLACE A MINUS SIGN IN FRONT OF THE
96 C   CHARACTER STRING BEFORE I SEND IT OUT.
97 C-----
98
99 110  IF(NEG.EQ.1)THEN         ! THE NUMBER IS NEGITIVE
100     NEG = 0                   ! RESET THE NEGATIVE INTEGER FLAG
101     CBUF = CNUM               ! MAKE A COPY OF THE CHARACTER STR
102     L = 1                      ! INITIALIZE CHARACTER PLACE CNTR
103
104     DO 120 I = 1, IN          ! LOOP THRU THE CHARACTER STRING
105     L = L + 1                 ! INCREMENT CHARACTER COUNTER

```

# Appendix IV

```

106      CNUM(L:L) - CBUR(L)  ! SHIFT THE STRING 1 PLACE RIGHT
107 120  CONTINUE
108
109      CNUM(1:1) - '-'      ! ADD THE MINUS SIGN TO FIRST CHR
110      IN - IN + 1          ! CORRECT THE CHARACTER COUNTER
111  ENDIF
112
113      LENGTH - IN
114
115  C   WRITE(5,' THE CHARACTER - ',CNUM,' IN - ',LENGTH
116  C
117  C   GOTO 5
118
119 1000  NUMBER - NUMBUF
120      RETURN
121      END

```

# Appendix IV

## AIV.11 Analog APSD Software Modules: INTRPT Source Code.

```

1      SUBROUTINE INTRPT(INT,CH)
2      C-----
3      C   THIS ROUTINE WILL TAKE AN INTEGER AND TURN IT INTO A
4      C   CHARACTER STRING EQUIVALENT TO THE INTEGER FOR THE
5      C   TEK TERMINALS.
6      C-----
7
8      CHARACTER CH*6,C
9      INTEGER INT,X1,X2
10     DIMENSION C(6)
11     ICNT=0
12     CH=' '
13     L=0
14     X1=MOD(INT,16)
15     X2=INT/16
16     IF(INT.GT.0) THEN
17         X1=X1+48
18     ELSE
19         X1=-X1+32
20     ENDIF
21     ICNT=ICNT+1
22     C(1)=CHAR(X1)
23     IF(INT.LE.0)GOTO 30
24     DO 20 I=2,6
25         IF (X2.GE.64)THEN
26             X1=MOD(X2,64)
27             X2=X2/64
28             X1=X1+64
29             C(I)=CHAR(X1)
30             ICNT=ICNT+1
31         ELSE
32             X1=X2+64
33             C(I)=CHAR(X1)
34             ICNT=ICNT+1
35             GOTO 30
36         ENDIF
37     20 CONTINUE
38     GOTO 30
39     30 DO 40 I=2,6
40         IF (X2.LE.-64)THEN
41             X1=MOD(X2,64)
42             X2=X2/64
43             X1=-X1+64
44             C(I)=CHAR(X1)
45             ICNT=ICNT+1
46         ELSE
47             X1=-X2+64
48             C(I)=CHAR(X1)
49             ICNT=ICNT+1
50             GOTO 30
51         ENDIF

```

# Appendix IV

```

32 40 CONTINUE
33 50 L-7
34      IP(NT.LT.16.AND.INT.GT.-16)ICNT-1
35      DO 60 J-1,ICNT
36          L-L-1
37          CH(L-L)-C(I)
38 60 CONTINUE
39      L-0
40      DO 70 I-1,6
41          IF(CH(I).EQ.' ')GOTO 70 ! IS THIS A BLANK SPACE
42          L-L+1
43          C(L)-CH(I)
44 70 CONTINUE          ! THESE DO LOOPS PLACE THE
45                      ! CHARACTERS ON THE LEFT
46      CH-' '
47      DO 80 I-1,L
48          CH(I)-C(I)
49 80 CONTINUE
50      L-0
51      RETURN
52      END

```

## Appendix IV

### AIV.12 Analog APSD Software Modules: LASER\_IN Source Code.

```

1  SUBROUTINE LASER_IN(SAMP,AGENT,CONC,NOSAMP,NAME,LAS,LAMDA,
2      LAS_ORD,TEXT)
3  C -----
4  C  THIS MOD IS USED FOR ENTERING SPECIFIC LASER DATA FOR EACH SAMPLE
5  C  IF THE USER WANTS TEKTRONIX GRAPHICS. THIS ROUTINE IS RUN JUST
6  C  AFTER THE SAMPLE INFORMATION GATHERING ROUTINE " TEK_INPUTS "
7  C -----
8
9
10  CHARACTER NAME*20,SAMP*20,AGENT*20,CONC*15,LAMDA*15
11  CHARACTER DATE*9,TIME*8,TEXT*80,E*1,A*5,SEG*3
12  CHARACTER CORRECT*2,SAMP CNT,LASERS,COUNT*3
13  CHARACTER SAMPLE*9,ERROR,L_ORDER
14
15  INTEGER CHANGE,ERRMSG,X,Y,X1,Y1,Y2,Y3
16  INTEGER LAS,LAS_ORD,ORDER
17
18  REAL REND,RINCR,RARM
19  REAL DIST
20
21  C -----
22
23  DIMENSION SAMP(10),AGENT(10),CONC(10)
24  DIMENSION LAS(10),LAMDA(40),LAS_ORD(40)
25  DIMENSION START(10),STOP(10),INC(10),POS(10)
26
27  DIMENSION COUNT(4)
28  DATA COUNT /'1st','2nd','3rd','4th'/
29
30  2  FORMAT(A20)
31  3  FORMAT(A4)
32  4  FORMAT('O')
33  5  FORMAT(A2)
34  6  FORMAT (A)
35  7  FORMAT(B)
36  8  FORMAT (A10)
37
38  E - CHAR(27)
39  CHANGE - 0
40  ISTART - 1
41  ICNT - 1          ! SAMPLE COUNTER
42  ISAMP - 1
43  IEXIT - 0
44
45  C -----
46
47  CALL INT_TO_CHAR(NOSAMP,SAMPLE,LE) ! CONVERT TO CHARACTER
48
49  C -----
50  C  THIS PART PLACES A RED PANEL SEGMENT 8000 NO THE ENTIRE SCREEN
51  C  FOR A BACKGROUND.

```

# Appendix IV

```

52  C-----
53
54  WRITE(*,"E//RUP"      ! PICKUP LEVEL 0
55  WRITE(*,"E//SK"       ! DELETE ALL SEGMENTS
56  WRITE(*,"E//RUP"      ! PICKUP LEVEL NORMAL
57
58  ISEG = 8000
59  CALL INTRPT (ISEG,SEG)
60  WRITE(*,"E//SE//SEG    ! BEGIN THE PANEL 8000
61  WRITE(*,"E//MP"       ! PANEL COLOR RED
62  X = 1
63  Y = 1
64  CALL HTY(X,Y,A)
65  WRITE(*,"E//LP//A     ! SET PANEL ORIGIN
66
67  X = 4095
68  CALL HTY(X,Y,A)
69  WRITE(*,"E//LG//A     ! DRAW BOTTOM OF PANEL
70  Y = 3276
71  CALL HTY(X,Y,A)
72  WRITE(*,"E//LG//A     ! DRAW LEFT SIDE OF PANEL
73  X = X - 4094
74  CALL HTY(X,Y,A)
75  WRITE(*,"E//LG//A     ! DRAW TOP OF PANEL
76  WRITE(*,"E//SC"       ! CLOSE AND FILL PANEL
77
78  C-----
79  C   THIS IS THE FIRST LINE THAT IS REQUIRED BY THE USER.
80  C   NAME, DATE, TIME AND NUMBER OF SAMPLES.
81  C-----
82
83  CALL TIME4(DATE,TIME)  ! SYSTEM TIME AND DATE
84
85  ISEG = 1
86  CALL INTRPT(ISEG,SEG)
87
88
89
90  WRITE(*,"E//SE//SEG    ! BEGIN THE SEGMENT
91  WRITE(*,"E//MT"       ! LINE COLOR WHITE
92  X = 100
93  Y = 3000
94  CALL HTY (X,Y,A)
95  WRITE(*,"E//LP//A     ! SET TEXT ORIGIN
96  WRITE(*,"E//LT//DATE   ! WRITE THE DATE
97  X = 1000
98  CALL HTY (X,Y,A)
99  WRITE(*,"E//LP//A     ! SET TEXT ORIGIN
100 WRITE(*,"E//LT//TIME   ! WRITE THE TIME
101
102 WRITE(*,"E//MT"       ! LINE COLOR RED
103 X = 2000
104 CALL HTY (X,Y,A)
105 WRITE(*,"E//LP//A     ! SET TEXT ORIGIN

```

# Appendix IV

```

106      TEXT - ' NAME'
107      WRITE(*,"E/LTY//TEXT      ! WRITE THE DATE
108      WRITE(*,"E/ISC"
109
110      C-----
111      C   THIS PART DRAWS A BOX AT THE BOTTOM OF THE SCREEN FOR THE DIALOG
112      C   AREA INPUTS.
113      C-----
114
115      ISEG - ISEG + 1
116      CALL INTRPT (ISEG,SEG)
117      WRITE(*,"E/SE//SEG      ! BEGIN THE PANEL 8000
118      WRITE(*,"E/ML1'      ! LINE COLOR WHITE
119      WRITE(*,"E/MP3'      ! PANEL COLOR BLUE
120      X - 1
121      Y - 1
122      CALL HTY(X,Y,A)
123      WRITE(*,"E/LP//A//Y'      ! SET PANEL ORIGIN
124      X - 4095
125      CALL HTY(X,Y,A)
126      WRITE(*,"E/FG//A      ! DRAW BOTTOM OF PANEL
127      Y - 450
128      CALL HTY(X,Y,A)
129      WRITE(*,"E/FG//A      ! DRAW LEFT SIDE OF PANEL
130      X - X - 4094
131      CALL HTY(X,Y,A)
132      WRITE(*,"E/FG//A      ! DRAW TOP OF PANEL
133      WRITE(*,"E/LE'      ! FILL THE PANEL
134
135      C-----
136      C   THIS DRAWS A LINE AROUND THE TWO DIALOG AREA LINES: COLOR RED
137      C-----
138
139      WRITE(*,"E/ML2'      ! PANEL COLOR RED
140      X - 100
141      Y - 150
142      CALL HTY(X,Y,A)
143      WRITE(*,"E/LP//A      ! SET PANEL ORIGIN
144      X - 3995
145      CALL HTY(X,Y,A)
146      WRITE(*,"E/FG//A      ! DRAW BOTTOM OF PANEL
147      Y - Y + 250
148      CALL HTY(X,Y,A)
149      WRITE(*,"E/FG//A      ! DRAW LEFT SIDE OF PANEL
150      X - 100
151      CALL HTY(X,Y,A)
152      WRITE(*,"E/FG//A      ! DRAW TOP OF PANEL
153      Y - Y - 250
154      CALL HTY(X,Y,A)
155      WRITE(*,"E/FG//A      ! DRAW LEFT SIDE OF PANEL
156
157      C-----
158
159      WRITE(*,"E/MT1'      ! TEXT COLOR WHITE

```

# Appendix IV

```

160
161 TEXT - 'ENTER < E > TO EXIT'
162
163 X - 1600
164 Y - 45
165 CALL HTY(X,Y,A)
166 WRITE(*,"E//LP//A" ! SET TEXT ORIGIN
167 WRITE(*,"E//LTAA//TEXT" ! WRITE THE TEXT-%y<n/Tnu
168 WRITE(*,"E//SC" ! CLOSE AND FILL PANEL
169
170 C-----
171 C THIS SETS UP THE DIALOG AREA SO THAT THE TEXT AND DATA ENTRY
172 C ARE ALL WITHIN THE DATA INPUT WINDOW.
173 C-----
174
175 WRITE(*,"E//LV0" ! DISABLE DIALOG AREA
176 WRITE(*,"E//LZ" ! CLEAR DIALOG AREA
177 WRITE(*,"E//LL2" ! DIALOG AREA 2 LINES
178 WRITE(*,"E//LCD1" ! 65 CHARACTER ALLOWED
179 WRITE(*,"E//ML1" ! DIALOG TEXT WHITE
180 X - 200
181 Y - 160
182 CALL HTY(X,Y,A)
183 WRITE(*,"E//LX//A" ! SET TEXT DIALOG ORIGIN
184
185
186
187 C-----
188 C THIS WRITES THE NAME OF THE PERSON RUNNING THE EXPERIMENT
189 C-----
190 C-----
191
192 ISEG - 5
193 CALL INTRPT(ISEG,SEG)
194
195 WRITE(*,"E//SK//SEG" ! DELETE THE SEGMENT
196
197 WRITE(*,"E//SE//SEG" ! BEGIN THE SEGMENT
198 WRITE(*,"E//MP/" ! PANEL COLOR GRAY
199 WRITE(*,"E//MT1" ! TEXT COLOR WHITE
200
201 X - 2300
202 Y - 2975
203 CALL HTY(X,Y,A)
204 WRITE(*,"E//LP//A" ! SET PANEL ORIGIN
205 X - X + 1100
206 CALL HTY(X,Y,A)
207 WRITE(*,"E//LC//A" ! DRAW BOTTOM OF PANEL
208 Y - Y + 100
209 CALL HTY(X,Y,A)
210 WRITE(*,"E//LC//A" ! DRAW LEFT SIDE OF PANEL
211 X - X - 1100
212 CALL HTY(X,Y,A)
213 WRITE(*,"E//LC//A" ! DRAW TOP OF PANEL

```

# Appendix IV

```

214      WRITE(*,"E/LE"      ! FILL THE PANEL
215
216      C-----
217      C   THIS PLACES THE TEXT IN THE PANEL
218      C-----
219
220      X - 2550
221      Y - 3000
222      CALL HTY(X,Y,A)
223      WRITE(*,"E/LP"//A      ! SET THE ORIGIN
224      WRITE(*,"E/LTA4"//NAME      ! WRITE THE NAME
225      WRITE(*,"E/SC"      ! CLOSE THE SEGMENT
226
227      C-----
228      C   THIS DRAWS THE SAMPLE DATA HEADER AND NUMBER OF SAMPLES
229      C-----
230
231      40   ISEG - 7
232      CALL INTRPT(ISEG,SEG)
233      WRITE(*,"E/PSK"//SEG      ! DELETE THE SEGMENT
234      WRITE(*,"E/PSE"//SEG      ! BEGIN THE SEGMENT
235
236      X - 1
237      Y - 2800
238      CALL HTY(X,Y,A)
239      WRITE(*,"E/MLA"      ! LINE COLOR BLUE
240      WRITE(*,"E/LP"//A      ! SET LINE ORIGIN
241
242      X - 4095
243      CALL HTY(X,Y,A)
244      WRITE(*,"E/PLG"//A      ! SET LINE END
245
246      WRITE(*,"E/MPY"      ! PANEL COLOR GRAY
247      WRITE(*,"E/MT1"      ! TEXT COLOR WHITE
248
249      X - 2500
250      Y - 2530
251      CALL HTY(X,Y,A)
252      WRITE(*,"E/LP"//A      ! SET PANEL ORIGIN
253
254      X - X + 1100
255      CALL HTY(X,Y,A)
256      WRITE(*,"E/PLG"//A      ! DRAW BOTTOM OF BOX
257
258      Y - Y + 100
259      CALL HTY(X,Y,A)
260      WRITE(*,"E/PLG"//A      ! RIGHT SIDE OF BOX
261
262      X - X - 1100
263      CALL HTY(X,Y,A)
264      WRITE(*,"E/PLG"//A      ! TOP OF BOX
265      WRITE(*,"E/LE"      ! FILL THE BOX
266
267      WRITE(*,"E/MT1"      ! TEXT COLOR WHITE

```

# Appendix IV

```

268
269 X = X + 100
270 Y = Y - 75
271 CALL HTY(X,Y,A)
272 WRITE(*,"E//LP//A          ! SET TEXT ORIGIN
273
274 CALL INT_TO_CHAR(KCNT,SAMP CNT,LE) ! CONVERT TO CHARACTER
275
276 CALL INT_TO_CHAR(NOSAMP,SAMPLE,LE) ! CONVERT TO CHARACTER
277 ! THE NUMBER OF SAMPLES
278
279 IF(NOSAMP.EQ.1)THEN
280 WRITE(*,"E//LT//1 OF 1 SAMPLES' ! WRITE THE TEXT
281 ELSE
282
283 WRITE(*,"E//LTA4//SAMP CNT// OF 'SAMPLE(1:LE)//
284 ' SAMPLES'
285
286 ENDIF
287
288 WRITE(*,"E//SC"          ! END SEGMENT
289
290 C-----
291 C THIS WRITE THE COLUMN HEADERS FOR " SAMPLE NAMES ", " AGENT TYPES "
292 C AND " CONCENTRATIONS "
293 C-----
294
295 ISEG = 8
296 CALL INTRPT(ISEG,SEG)
297 WRITE(*,"E//SK//SEG
298 WRITE(*,"E//SE//SEG          ! BEGIN THE SEGMENT
299
300 WRITE(*,"E//MT"          ! TEXT COLOR YELLOW
301 X = 500
302 Y = 2250
303 CALL HTY(X,Y,A)
304 WRITE(*,"E//LP//A          ! SET TEXT ORIGIN
305 TEXT = 'SAMPLE:'
306 WRITE(*,"E//LT//TEXT          ! WRITE SAMPLE TEXT
307
308 X = 1800
309 CALL HTY(X,Y,A)
310 WRITE(*,"E//LP//A          ! SET TEXT ORIGIN
311 TEXT = 'AGENT:'
312 WRITE(*,"E//LT>//TEXT          ! WRITE SAMPLE TEXT
313
314 X = 3000
315 CALL HTY(X,Y,A)
316 WRITE(*,"E//LP//A          ! SET TEXT ORIGIN
317 TEXT = 'CONCENTRATION'
318 WRITE(*,"E//LTA2//TEXT          ! WRITE SAMPLE TEXT
319
320 WRITE(*,"E//SC"
321

```

# Appendix IV

```

322 C-----
323
324 50 ISEG - 10
325 CALL INTRPT(ISEG,SEG)
326 WRITE(*,"E/PSK"/ISEG
327 WRITE(*,"E/PSE"/ISEG
328
329 Y - 2100
330 X - 200
331 CALL HTY(X,Y,A)
332 WRITE(*,"E/PLP"/A ! SET PANEL ORIGIN
333
334 X - X + 1100
335 CALL HTY(X,Y,A)
336 WRITE(*,"E/LG"/A ! DRAW BOTTOM OF BOX
337
338 Y - Y + 100
339 CALL HTY(X,Y,A)
340 WRITE(*,"E/LG"/A ! RIGHT SIDE OF BOX
341
342 X - X - 1100
343 CALL HTY(X,Y,A)
344 WRITE(*,"E/LG"/A ! TOP OF BOX
345 WRITE(*,"E/LE" ! FILL THE BOX
346
347 WRITE(*,"E/MT1" ! TEXT COLOR WHITE
348
349 X - X + 100
350 Y - Y - 75
351 CALL HTY(X,Y,A)
352 WRITE(*,"E/PLP"/A ! SET TEXT ORIGIN
353
354 WRITE(*,"E/LTY"/SAMP(ICNT) ! WRITE THE SAMPLE.
355 Y - Y - 25
356
357 C-----
358
359 X - 1300
360 CALL HTY(X,Y,A)
361 WRITE(*,"E/PLP"/A ! SET PANEL ORIGIN
362
363 X - X + 1100
364 CALL HTY(X,Y,A)
365 WRITE(*,"E/LG"/A ! DRAW BOTTOM OF BOX
366
367 Y - Y + 100
368 CALL HTY(X,Y,A)
369 WRITE(*,"E/LG"/A ! RIGHT SIDE OF BOX
370
371 X - X - 1100
372 CALL HTY(X,Y,A)
373 WRITE(*,"E/LG"/A ! TOP OF BOX
374 WRITE(*,"E/LE" ! FILL THE BOX
375

```

# Appendix IV

```

376      WRITE(*,"E//MT1"          ! TEXT COLOR WHITE
377
378      X - X + 100
379      Y - Y - 75
380      CALL HTY(X,Y,A)
381      WRITE(*,"E//LP//A          ! SET TEXT ORIGIN
382      WRITE(*,"E//LT//AGENT(ICNT)
383      Y - Y - 25
384
385      C _____
386
387      X - 2800
388      CALL HTY(X,Y,A)
389      WRITE(*,"E//LP//A          ! SET PANEL ORIGIN
390
391      X - X + 1100
392      CALL HTY(X,Y,A)
393      WRITE(*,"E//LG//A          ! DRAW BOTTOM OF BOX
394
395      Y - Y + 100
396      CALL HTY(X,Y,A)
397      WRITE(*,"E//LG//A          ! RIGHT SIDE OF BOX
398
399      X - X - 1100
400      CALL HTY(X,Y,A)
401      WRITE(*,"E//LG//A          ! TOP OF BOX
402      WRITE(*,"E//LE"            ! FILL THE BOX
403
404      WRITE(*,"E//MT1"          ! TEXT COLOR WHITE
405
406      X - X + 100
407      Y - Y - 75
408      CALL HTY(X,Y,A)
409      WRITE(*,"E//LP//A          ! SET TEXT ORIGIN
410      WRITE(*,"E//LT//CONC(ICNT)
411      WRITE(*,"E//SC"
412
413      C _____
414      C   THIS NEXT PART GETS THE NUMBER OF AGENTS...
415      C _____
416
417      52  FORMAT(SX,"ENTER THE NUMBER OF LASERS ( 1 - 4 ) : ",$)
418
419      WRITE(*,"E//LV1"          ! ENABLE DIALOG AREA
420
421      60  WRITE(*,"E//LZ"        ! CLEAR THE DIALOG AREA
422      WRITE(*,$2)
423
424      READ(*,6)LASERS           ! READ LASER COUNT
425
426      IF(LASERS.EQ.'E'.OR.LASERS.EQ.'4')THEN ! USER TO EXIT
427      EXIT - 1                  ! SET EXIT FLAG
428      GOTO 1000                 ! RETURN TO CALLER
429      ENDIF

```

# Appendix IV

```

430
431 READ(LASERS,(BN,D),ERR=60)LCNT ! CONVERT TO INTEGER
432
433 IF(LCNT.LT.1.OR.LCNT.GT.4)GOTO 60 ! BAD INPUT DO IT AGAIN
434
435 C-----
436 C THIS BEGINS THE LASER INPUT SECTION...
437 C-----
438
439 70 ISEG - 19
440
441 X - 350
442 Y - 1750
443
444 CALL INTRPT(ISEG,SEG)
445 WRITE(*,*)E//SK//SEG ! DELETE THE SEGMENT
446 WRITE(*,*)E//SE//SEG ! BEGIN THE SEGMENT
447
448 WRITE(*,*)E//ML1' ! LINE COLOR WHITE
449 WRITE(*,*)E//MP# ! PANEL COLOR GREEN
450 WRITE(*,*)E//MT' ! TEXT COLOR YELLOW
451
452
453 CALL HTY(X,Y,A)
454 WRITE(*,*)E//LP//A ! SET TEXT ORIGIN
455 TEXT = '0.'
456 WRITE(*,*)E//LT2//TEXT ! WRITE THE TXT
457
458 WRITE(*,*)E//MT4' ! TEXT COLOR BLUE
459 X - X + 150
460
461 CALL HTY(X,Y,A)
462 WRITE(*,*)E//LP//A//1' ! SET PANEL ORIGIN
463 X - X + 700
464 CALL HTY(X,Y,A)
465 WRITE(*,*)E//LG//A ! DRAW BOTTOM OF BOX
466
467 Y - Y + 100
468 CALL HTY(X,Y,A)
469 WRITE(*,*)E//LG//A ! RIGHT SIDE OF BOX
470
471 X - X - 700
472 CALL HTY(X,Y,A)
473 WRITE(*,*)E//LG//A ! TOP OF BOX
474 WRITE(*,*)E//LE' ! FILL THE BOX
475
476 Y - Y - 75
477 X - X + 150
478 CALL HTY(X,Y,A)
479 WRITE(*,*)E//LP//A ! SET TEXT ORIGIN
480 CALL INT_TO_CHAR(LCNT,LASERS,LE)
481 WRITE(*,*)E//LT//LASERS//LASERS'
482 WRITE(*,*)E//SC' ! CLOSE THE SEGMENT
483

```

# Appendix IV

```

484
485 C-----
486 C   THE WRITES THE TEXT HEADER INFORMATION. THE NUMBER OF THE ITEM
487 C   FOR THE LASER ORDER AND WAVELENGTH.
488 C-----
489
490     ISEG - 21
491     CALL INTRPT(ISEG,SEG)
492     WRITE(*,"E//SK//SEG
493     WRITE(*,"E//SE//SEG
494     WRITE(*,"E//MT7"           ! TEXT COLOR YELLOW
495     X - 2350
496     Y - 1820
497     CALL HTY(X,Y,A)
498     WRITE(*,"E//LP//A"         ! SET TEXT ORIGIN
499     TEXT - 'LASER'
500     WRITE(*,"E//LTS//TEXT"     ! WRITE THE TXT
501     Y - 1750
502     CALL HTY(X,Y,A)
503     WRITE(*,"E//LP//A"         ! SET TEXT ORIGIN
504     TEXT - 'ORDER'
505     WRITE(*,"E//LTS//TEXT"     ! WRITE THE TXT
506
507     X - 3050
508     Y - 1750
509     CALL HTY(X,Y,A)
510     WRITE(*,"E//LP//A"         ! SET TEXT ORIGIN
511     TEXT - 'WAVELENGTH'
512     WRITE(*,"E//LT//TEXT"      ! WRITE THE TXT
513     WRITE(*,"E//SC"
514
515
516 C-----
517 C   THIS BEGINS A LOOP THRU THE LASERS FOR EACH SAMPLE. THE LASER
518 C   NUMBER ( 1 - 4 ) IS TAKEN ALONG WITH THE WAVELENGTH THE LASER WILL
519 C   BE TUNED TOO.
520 C-----
521
522     WRITE(*,"E//LZ"           ! CLEAR DIALOG AREA
523     LAS(ICNT) - LCNT          ! PLACE NUMBER OF LASERS
524
525
526
527     DO 400 I = 1, LCNT        ! LOOP THRU THE LASERS
528
529 C-----
530 C   THIS DRAWS THE PANEL THAT SHOWS WHICH LASER IS TO BE DESCRIBED.
531 C-----
532
533
534     Y1 - 1200
535     X - 500
536     Y - Y1
537

```

# Appendix IV

```

538     ISEG - 20
539     CALL INTRPT(ISEG,SEG)
540     WRITE(*,"E/PSK"/ISEG      ! DELETE THE SEGMENT
541     WRITE(*,"E/SE"/ISEG      ! BEGIN THE SEGMENT
542
543     CALL HTY(X,Y,A)
544     WRITE(*,"E/PLP"/A/P1'      ! SET PANEL ORIGIN
545     X - X + 1000
546     CALL HTY(X,Y,A)
547     WRITE(*,"E/PLG"/A      ! DRAW BOTTOM OF BOX
548
549     Y - Y + 100
550     CALL HTY(X,Y,A)
551     WRITE(*,"E/PLG"/A      ! RIGHT SIDE OF BOX
552
553     X - X - 1000
554     CALL HTY(X,Y,A)
555     WRITE(*,"E/PLG"/A      ! TOP OF BOX
556     WRITE(*,"E/PLE'      ! FILL THE BOX
557
558     Y - Y - 75
559     X - X + 100
560     CALL HTY(X,Y,A)
561     WRITE(*,"E/PLP"/A      ! SET TEXT ORIGIN
562
563     WRITE(*,"E/PMT4'      ! TEXT COLOR BLUE
564
565     IF(LASERS.EQ.'1')THEN
566     WRITE(*,"E/LTP"/1 OF 1 LASER'      ! WRITE THE TEXT
567
568     ELSE
569     CALL INT_TO_CHAR(LCNT,LASERS,LE)
570     WRITE(*,"E/LTP"/COUNT(I)/' OF 'LASERS/' LASERS'
571
572     ENDIF
573
574     WRITE(*,"E/SC'      ! CLOSE THE SEGMENT
575
576     C-----
577
578     100  FORMAT(SX,'ENTER 'A3,' LASER NUMBER: ',S)
579
580     102  FORMAT(SX,'ENTER THE LASER NUMBER: (1 - 4) ',S)
581
582
583     110  IF(LCNT.EQ.1)THEN      ! IF ONLY ONE LASER THEN
584     WRITE(*,102)      ! GET THE LASER NUMBER
585
586     ELSE
587     WRITE(*,100)COUNT(I)
588
589     ENDIF
590
591     READ (*,A,ERR=110)LCNT      ! READ THE LASER ORDER

```

# Appendix IV

```

592
593      IF(L_ORDER.EQ.'')GOTO 110      ! LOOK FOR NULL
594
595      IF(L_ORDER.EQ.'E'.OR.L_ORDER.EQ.'e')THEN ! USER TO EXIT
596          EXIT = 1                ! SET EXIT FLAG
597          GOTO 1000                ! RETURN TO CALLER
598      ENDIF
599
600      READ(L_ORDER,'(BN,I)',ERR=110)ORDER ! MAKE CHAR AN INTEGER
601
602      IF(ORDER.LT.1.OR.ORDER.GT.4)GOTO 110 ! EVALUATE BOUNDS
603
604      LAS_ORD(I + MCNT) = ORDER      ! PLACE IN ARRAY
605
606      C----- WRITE(*,"E//RP0"      ! FIXUP LEVEL 0
607      WRITE(*,"E//SK"                ! DELETE ALL SEGMENTS
608      WRITE(*,"E//RP6"                ! FIXUP LEVEL NORMAL
609
610      RETURN
611      END

```

## Appendix IV

### AIV.13 Analog APSD Software Modules: MENU Source Code.

```

1      SUBROUTINE MENU
2      C-----
3      C   THIS IS A MENU FOR THE MULLER MATRIX EXPERIMENT FOR USERS THAT
4      C   DO NOT HAVE A GRAPHICS TERMINAL.
5      C-----
6
7      CHARACTER GOOP,DUMZ*5
8      CHARACTER *40 TITLE,T1,T2,T3,T4,T5
9
10
11     TITLE=' MULLER MATRIX EXPERIMENT'
12     T1='1. BEGIN A NEW EXPERIMENT'
13     T2='2. REVIEW COLLECTED DATA'
14     T3='3. CALIBRATE OPTIC STAGES'
15     T4='4. CALIBRATE A/D CONVERTER'
16     T5='5. EXIT'
17
18     10  PRINT 1,TITLE           ! THIS PRINTS THE ABOVE MENU
19     PRINT 2,T1
20     PRINT 2,T2
21     PRINT 2,T3
22     PRINT 2,T4
23     PRINT 3,T5
24
25     1  FORMAT('1',10/),T1,59(' ')2(T1,~,T69,~/)
26     $ T1,~,T21,A,T69,~/T1,~,T69,~/T1,~,T69,~/
27     2  FORMAT(T1,~,T21,A,T69,~/T1,~,T69,~/T1,~,T69,~/)
28     3  FORMAT(T1,~,T21,A,T69,~/T1,~,T69,~/T1,~,T69,~/)
29     $ T1,59(' ')
30     4  FORMAT (A)
31
32
33     C-----
34     C   THIS ROUTES THE USER TO THE PROPER SUBROUTINE SELECTION
35     C-----
36
37     READ (*,*,ERR=600) I
38
39     IF (I.LT.1.OR.I.GT.5) GOTO 600
40
41     GOTO (100,200,300,400,500) I
42
43     100  CALL NEW_TEX(0)         ! BEGIN A NEW EXP
44     GOTO 10
45
46     200  CALL LOOKUP             ! LOOK AT OLD DATA
47     GOTO 10
48
49     300  CALL STAGE_POSITION      ! CALIBRATE OPTIC STAGES
50     GOTO 10
51

```

## Appendix IV

```
52 400 CALL A2D_CAL      ! CALIBRATE DATA COLLECTION
53     GOTO 10
54
55 600 WRITE (*,*)'PLEASE KEEP YOUR NUMBERS WITHIN THE LIST'
56     WRITE (*,*) 'RETURN TO CONTINUE'
57     READ (*,4) GOOF
58     GOTO 10
59
60
61
62 500 RETURN          ! USER WANTS TO EXIT
63     END
```

## Appendix IV

### *AIV.14 Analog APSD Software Modules: MIX Source Code.*

```

1      SUBROUTINE MIX(A,NUMBER)
2      C -----
3      C   THIS ROUTINE CHANGES INTEGERS OF 9999 OR LESS INTO CHARACTERS
4      C   SO THAT THEY CAN BE USED AS NAMES OF FILES FOR THE SAVED DATA
5      C   STRINGS.
6      C
7      C
8
9      CHARACTER NUMBER*5
10     INTEGER  A,HTHOU,THOU,HUND,TEN,ONES
11     C   READ(*,*)A
12     C
13     C   THIS WILL SUPPRESS LEADING ZEROS TO THE CHARACTER STRING
14     C
15     BUFP=A
16     IF (A.GE.10000)THEN
17         NUMLEN=1
18         GOTO 2
19     ELSEIF (A.GE.1000)THEN
20         NUMLEN=2
21         GOTO 5
22     ELSE IF (A.GE.100) THEN
23         NUMLEN=3
24         GOTO 17
25     ELSE IF (A.GE.10) THEN
26         NUMLEN=4
27         GOTO 35
28     ELSE
29         NUMLEN=5
30         GOTO 55
31     END IF
32     C
33     C   THIS PART SEPARATES THE INTEGER INTO SINGLE PLACE VALUES
34     C
35     2   DO 3 I=90000,0,-10000
36         J=I-A
37         K=K+1
38         IF (J.LE.0) GOTO 4
39     3   CONTINUE
40
41
42     4   HTHOU=(10-K)
43     C   WRITE(*,*)HTHOU- ',HTHOU
44     A=A-(HTHOU*10000)
45     K=0
46
47     5   DO 10 I=9000,0000,-1000
48         J=I-A
49         K=K+1
50         IF (J.LE.0) GOTO 15
51     10  CONTINUE

```

# Appendix IV

```

52
53
54 15 THOU-(10-K)
55 C WRITE(,"YTHOU- ",THOU
56 A-A-(THOU*1000)
57 17 K-0
58 DO 20 I=900,0,-100
59 J-I-A
60 K-K+1
61 IF(J.LE.0) GOTO 30
62 20 CONTINUE
63
64 30 HUND-(10-K)
65 C WRITE(,"YHUND- ",HUND
66 A-A-(HUND*100)
67 35 K-0
68 DO 40 I=90,0,-10
69 J-I-A
70 K-K+1
71 IF(J.LE.0) GOTO 50
72 40 CONTINUE
73
74 50 TEN-(10-K)
75 C WRITE(,"YTEN- ",TEN
76 A-A-(TEN*10)
77 55 K-0
78 DO 60 I=9,0,-1
79 J-I-A
80 K-K+1
81 IF(J.LE.0) GOTO 70
82 60 CONTINUE
83
84 70 ONES-(10-K)
85 C WRITE(,"YONES- ",ONES
86
87 C
88 C HERE THE VALUE OF THE INTEGER IS CHANGED TO A CHARACTER
89 C
90
91 GOTO (90,100,200,300,400) NUMLEN
92
93 90 NUMBER=CHAR(HITHOU+48)/CHAR(THOU+48)
94 $ //CHAR(HUND+48)/CHAR(TEN+48)/CHAR(ONES+48)
95 GOTO 300
96
97 100 NUMBER=CHAR(THOU+48)/CHAR(HUND+48)/CHAR(TEN+48)
98 $ //CHAR(ONES+48)
99 GOTO 300
100 200 NUMBER=CHAR(HUND+48)/CHAR(TEN+48)/CHAR(ONES+48)
101 GOTO 300
102 300 NUMBER=CHAR(TEN+48)/CHAR(ONES+48)
103 GOTO 300
104
105 400 NUMBER=CHAR(ONES+48)

```

## Appendix IV

106  
107 500 A-BUFF  
108 K-0  
109 RETURN  
110 END

# Appendix IV

## AIV.15 Analog APSD Software Modules: MOV\_STAGE Source Code.

```

1      SUBROUTINE MOV_STAGE(LASER_NUM,TYPE,LAMDA,START_POS,REPORT,
2          GRAPHICS)
3      C
4      C-----
5      C   THIS ROUTINE MOVES THE OPTIC STAGES TO THEIR PROPER POSITIONS FOR
6      C   CHANNELING THE LASER BEAM TO EITHER THE SPECTRUM ANALYSER OR THE
7      C   SAMPLE.
8      C
9      C   LASER_NUM -   THIS IS THE USER DEFINED LASER NUMBER FOR WHICH
10     C               THE MOVEMENT OF THE OPTIC STAGES IS BEING
11     C               REQUESTED.
12     C
13     C   TYPE      -   1. USER IS REQUESTING THAT THE LASER BEAM BE
14     C               DIRECTED TO THE SPECTRUM ANALYZER.
15     C
16     C               2. USER IS REQUESTING THAT THE LASER BEAM BE
17     C               DIRECTED AT THE SAMPLE.
18     C
19     C               3. USER IS REQUESTING THAT THE SAMPLE STAGE BE
20     C               SENT TO ITS START POSITION.
21     C
22     C   LAMDA      -   WAVELENGTH OF THE LASER BEING TUNED.
23     C
24     C   START_POS -   THE DISTANCE IN STEPS FOR AXIS 1 TO MOVE THE
25     C               THE SAMPLE STAGE TO ITS HOME POSITION.
26     C
27     C   RECORD     -   THIS IS AN ERROR FLAG SENT BACK TO THE USER
28     C               IF ANY ERROR CONDITION OCCURS DURING THIS
29     C               STAGE MOVEMENT OPERATION.
30     C
31     C   GRAPHICS   -   INTEGER FLAG INDICATING THAT A TEKTRONIX TERMINAL
32     C               IS IN USE AND TO CALL THE GRAPHICS ROUTINE
33     C               TEK_TEXT.
34     C
35     C               ----
36     C-----
37
38     CHARACTER*10 VEL,ACC,DIST,DUMMY,START_POS,PORT
39     CHARACTER*1 LASER,CS,AXIS,CR,INPUT,ANS
40     CHARACTER*20 FILENM
41     CHARACTER*80 MSG,MSG1
42     CHARACTER*15 LAMDA,ANALYTE,AMOUNT
43
44     INTEGER TYPE,GRAPHICS,TXT_FLG,RESET
45
46     LOGICAL T
47
48     DIMENSION VEL(4),ACC(4),DIST(4),DUMMY(4)
49
50     C-----
51     C   THIS CHECKS TO SEE IF THE TYPE - 3. THIS IS THE CALLER ASKING

```

# Appendix IV

```

32      C   THAT THE SAMPLE STAGE BE MOVED TO ITS USER DEFINED START POSITION.
33      C_____
34
35      IF(TYPE.EQ.3)THEN
36          VEL(1:2) - '10'
37          ACC(1:2) - '10'
38          DIST(1) - START_POS
39          GOTO 50
40      ENDIF
41
42      C_____
43
44      5   FORMAT(12(A10))
45      ICNT - 0
46      CR - CHAR(13)      ! CARRIAGE RETURN
47      GS - CHAR(29)      ! CHARACTER REQUIRED TO ESTABLISH
48                          ! A SERIAL THRU LINK TO A DEVICE
49      L - LASER_NUM      ! TRANSFER THE LASER NUMBER TO "L"
50      REPORT - 0         ! INITIALIZE ERROR FLAG TO "NONE"
51
52      C_____
53      C   THIS PART ESTABLISHES THE DATA FILE NAME THAT HOLDS THE STAGE
54      C   MOVEMENT DATA FOR EACH LASER.
55      C_____
56
57      IF(TYPE.EQ.1)THEN      ! USER WANTS TO CHANNEL BEAM TO
58          FILENM - 'SPECTRUM' ! THE SPECTRUM ANALYZER
59
60      ELSEIF(TYPE.EQ.2)THEN  ! OR ELSE THE USER WANTS TO
61          FILENM - 'SAMP'    ! CHANNEL THE BEAM TO THE SAMPLE
62
63      ELSE                   ! OR ELSE INPUT WAS BAD SET
64          REPORT - 1         ! THE ERROR FLAG AND RETURN
65          GOTO 1000          ! TO THE USER
66      ENDIF
67
68      C_____
69      C   THIS INQUIRES IF THE DATA FILE EXISTS PRIOR TO OPENING THE FILE
70      C   IF IT DOES NOT THEN THE REPORT FLAG IS SET THE ROUTINE IS ENDED
71      C_____
72
73      INQUIRE(FILE=FILENM,EXIST = T) ! ASK IF FILE EXISTS
74
75      IF(.NOT.T)THEN         ! IF THERE IS NO FILE
76          REPORT - 2         ! SET THE ERROR FLAG
77          GOTO 1000          ! AND RETURN TO THE
78          ENDIF              ! CALLER
79
80      C_____
81      C   HERE THE PROPER FILE IS OPENED.
82      C_____
83
84      OPEN(2,FILE = FILENM,ACCESS='DIRECT',FORM='FORMATTED',
85      .RECL=120,STATUS = 'OLD',SHARED,ERR = 20)

```

# Appendix IV

```

106
107     GOTO 30          ! OPEN WAS SUCCESSFUL MOVE ON
108
109 20  REPORT - 3      ! OPEN FAILURE SET THE REPORT FLAG
110     GOTO 1000       ! RETURN TO CALLER
111
112 C _____
113 C   THIS PART READS THE SPECIFIED RECORD FOR THE LASER PASSED IN
114 C   AS THE VARIABLE LASER_NUM. THE NUMBER IS THE FILE RECORD NUMBER.
115 C _____
116
117
118 30  READ(2,REC - 1,PMT - 5,ERR - 40)VEL(1),ACC(1),DIST(1),
119     .DUMMY(1),VEL(2),ACC(2),DIST(2),DUMMY(2),VEL(3),ACC(3),
120     .DIST(3),DUMMY(3)
121
122     GOTO 50          ! THE READ WAS GOOD MOVE ALONG
123
124 40  REPORT - 4      ! THE READ FAILED. SET THE FLAG AND
125     GOTO 1000
126
127 C _____
128 C   HERE THE STEPPER MOTORS ARE MOVED FROM THEIR HOME POSITION THE
129 C   VELOCITY, ACCELERATION AND DISTANCE SPECIFIED BY THE ABOVE INPUT.
130 C   THIS IS DONE BY USE OF A DO LOOP THAT LOOPS THRU ALL THREE AXIS
131 C   ON THE CONTROLLER.
132 C _____
133
134 50  DO 130 I = 1,3
135
136 C _____
137 C   FIRST I MAKE SURE THAT THERE IS DATA PRIOR TO WRITING TO THE PORT
138 C   NO VELOCITY OR ACCELERATION MEANS NO MOVEMENT. THE ROUTINE THAT
139 C   CREATES THESE MOVEMENT FILES UNDERSTANDS THAT NOT ALL AXIS OR
140 C   STAGES ARE REQUIRED TO MOVE EVERY TIME AND THUS THE RECORDS
141 C   ARE FILLED WITH ZEROS.
142 C _____
143
144     IF(VEL(I).EQ.'0'.OR.ACC(I).EQ.'0'.OR.DIST(I).EQ.'0')THEN
145         ICNT = ICNT + 1
146         GOTO 130
147     ENDIF
148
149 C _____
150 C   THIS TURNS ON THE POWER TO THE STEPPER MOTORS AS THEY ARE NEEDED
151 C   THE AXIS SPECIFIES THE SPECIFIC MOTOR.
152 C _____
153
154     AXIS = CHAR(ICNT + 1)      ! CHANGE INTEGER TO CHARA
155
156     WRITE(3,'(A5)AXIS/ST1 '   ! TURN STEPPER POWER ON
157
158 C _____
159 C   THIS CREATES THE COMMAND LINE THAT IS SENT TO THE CONTROLLER

```

# Appendix IV

```

160 C IT IS A CHARACTER STRING THAT MUST BE EXACT IN ITS LENGTH. THIS
161 C IS SO BECAUSE THE CONTROLLER BUFFERS ITS DATA AND DOES NOT LIKE
162 C MANY SPACES ON THE END OF THE COMMAND LINE. AFTER MUCH TRIAL
163 C AN ERROR I HAVE FOUND THAT THER IS A POSSIBILITY OF 13 FORMATS
164 C OR CHARACTER LENGTHS THAT MAY BE REQUESTED OF THIS ROUTINE. I
165 C PROVIDE THEM HERE.
166 C THE COMMAND LINE IS ALSO A CHARACTER STRING.
167 C
168 C AXIS - THE STEPPER MOTOR CONTROLLER AXIS NUMBER
169 C MSG1 - THE COMMAND LINE THAT TELLS THE VELOCITY,
170 C ACCELERATION, AND DISTANCE FOR EACH AXIS.
171 C _____
172
173 450 FORMAT(A11)
174 451 FORMAT(A12)
175 452 FORMAT(A13)
176 453 FORMAT(A14)
177 454 FORMAT(A15)
178 455 FORMAT(A16)
179 456 FORMAT(A17)
180 457 FORMAT(A18)
181 458 FORMAT(A19)
182 459 FORMAT(A20)
183 460 FORMAT(A21)
184 461 FORMAT(A22)
185 462 FORMAT(A23)
186
187 C _____
188
189
190 MSG - AXIS//V//VEL(I) ! BEGIN COMMAND LINE W/VEL
191 MSG1 - ''
192
193 M - 0 ! M - CHARACTER COUNTER
194
195 DO 70 K - 1,3 ! LOOP THRU EACH SETTING
196
197 DO 60 J - 1,12 ! LOOP THRU EACH CHARACTER
198
199 IF(ICHAR(MSG(J)))LT.49)GOTO 60
200 ! DONT COUNT SPACES OR "+"
201 M - M + 1 ! COUNT STRING LENGTH
202 MSG1(M:M) - MSG(J:J) ! BUILD THE STRING
203 60 CONTINUE
204
205 M - M + 1 ! INCREMENT STRING COUNT
206 MSG1(M:M) - '' ! ADD A SPACE AFTER
207 ! THE COMMAND.
208 IF(K.EQ.1)MSG - AXIS//A//ACC(I) ! ADD ACCELERATION
209 IF(K.EQ.2)MSG - AXIS//D//DIST(I) ! ADD TRAVEL DISTANCE
210
211 70 CONTINUE
212
213 M - M + 1

```

# Appendix IV

```

214      MSG1(M:M + 1) - 'G'      ! ADD A GO TO THE END
215      M - M + 1
216      J - M - 10
217
218      CALL TWAIT(5)
219
220      GOTO(570,580,590,600,610,620,630,640,650,660,670,680,690)
221
222 570  WRITE(3,450)MSG1(1:M)
223      GOTO 100
224 580  WRITE(3,451)MSG1(1:M)
225      GOTO 100
226 590  WRITE(3,452)MSG1(1:M)
227      GOTO 100
228 600  WRITE(3,453)MSG1(1:M)
229      GOTO 100
230 610  WRITE(3,454)MSG1(1:M)
231      GOTO 100
232 620  WRITE(3,455)MSG1(1:M)
233      GOTO 100
234 630  WRITE(3,456)MSG1(1:M)
235      GOTO 100
236 640  WRITE(3,457)MSG1(1:M)
237      GOTO 100
238 650  WRITE(3,458)MSG1(1:M)
239      GOTO 100
240 660  WRITE(3,459)MSG1(1:M)
241      GOTO 100
242 670  WRITE(3,460)MSG1(1:M)
243      GOTO 100
244 680  WRITE(3,461)MSG1(1:M)
245      GOTO 100
246 690  WRITE(3,462)MSG1(1:M)
247
248 100  IF(TYPE.EQ.3)GOTO 1000      ! EXIT IF MOVING STAGE TO START
249
250 130  CONTINUE
251
252  C _____
253  C   AT THIS POINT THE ROUTINE MAKES A SWITCH BASED ON THE TYPE OF
254  C   DATA REQUESTED BY THE CALLER.
255  C
256  C   TYPE - 1 THE SPECTRUM ANALYZER IS THE TARGET. THE STAGES
257  C   ARE NOW MOVED TO THE CORRECT LOCATON. THE USER
258  C   IS REQUESTED TO PRESS RETURN WHEN THE LASER
259  C   TUNNING IS COMPLETE. THE STAGES ARE RETURNED TO
260  C   THEIR HOME POSITIONS AND THE NEXT LASER LOOPS IN
261  C   TURN. < IF THERE ARE MORE LASERS >
262  C
263  C   TYPE - 2 THE SAMPLE IS THE TARGET. SINCE THE STAGES ARE
264  C   NOW IN THE CORRECT POSITION FOR DATA COLLECTION
265  C   THE ROUTINE IS EXITED. THE STAGES ARE SENT HOME
266  C   BY THE CALLER AFTER DATA COLLECTION.
267  C _____

```

# Appendix IV

```

268
269      IF(TYPE.EQ.1)THEN          ! BEAM IS SPECTRUM ANALYZER
270 108  FORMAT(24//)
271 110  FORMAT(20X,'LASER : ',I4,20X,'WAVELENGTH : ',A15,/)
272 112  FORMAT(10X,'PRESS RETURN WHEN THE LASER IS CALIBRATED:',/)
273 114  FORMAT(35X,'WAITING...',/)
274 116  FORMAT(40X,$)
275
276  C-----
277  C   THIS PART CALLS THE GRAPHICS ROUTINE TO DISPLAY THE TEXT ABOVE.
278  C-----
279
280      IF(GRAPHICS.EQ.1)THEN
281          RESET - 1              ! DO NOT REDRAW THE PANEL.
282          ANALYTE - LAMDA        ! PLACE WAVELENGTH AND
283          ANS - CHAR(L + 48)     ! LASER No. IN VARIABLES
284          TXT_FLG - 4           ! FLAG FOR ABOVE TEXT
285          CALL TEK_TEXT(TXT_FLG,PORT,RESET,ANS,ANALYTE,AMOUNT,TEXT)
286
287          GOTO 140
288      ENDIF
289  C-----
290
291 120  WRITE(*,108)
292      WRITE(*,110)L,LAMDA
293      WRITE(*,112)
294      WRITE(*,114)
295      WRITE(*,116)
296      READ(*,*(A),ERR=120)INPUT
297
298  C-----
299  C   REVERSE DIRECTION AND < G > TELLS IT TO GO THE SAME DISTANCE
300  C   AND VELOCITY. ICNT KEPT A COUNT ON THE NUMBER OF STAGES NOT
301  C   MOVED DURING THIS OPERATION. IF ITS - 3 THEN THERE IS NO HOME TO
302  C   GO TOO.
303  C-----
304
305 140  IF(ICNT.LT.3)WRITE(3,*(A4))'H G ' ! SEND ALL THE STAGES HOME
306
307
308      ENDIF
309
310  C-----
311  C   THIS PROVIDES THE USER WITH ERROR INFORMATION IN THE UNLIKELY
312  C   EVENT THAT A PROBLEM OCCURS WITH THIS ROUTINE.
313  C   THIS IS BASED ON THE NUMBER OF THE " REPORT " VARIABLE.
314  C-----
315
316 1000  IF(REPORT.GT.0)THEN        ! THERE IS A PROBLEM
317
318 1010  FORMAT(10X,'NO FILE TYPE WAS PROVIDED BY THE CALLER',/)
319 1020  FORMAT(10X,'THE FILE: ',A20,' CANNOT BE POUND',/)
320 1030  FORMAT(10X,'THERE WAS AN OPEN ERROR ON THE FILE: ',A20,/)
321 1040  FORMAT(10X,'THE FILE: ',A20,' FAILED ON THE READ',/)

```

## Appendix IV

```
322
323 IF(REPORT.EQ.1)WRITE(*,1010)
324 IF(REPORT.EQ.2)WRITE(*,1020)FILENM
325 IF(REPORT.EQ.3)WRITE(*,1030)FILENM
326 IF(REPORT.EQ.4)WRITE(*,1040)FILENM
327
328 ENDIF
329
330 WRITE(3, '(A4)') 'ST0'      ! TURN STEPPER POWER OFF
331
332 RETURN                    ! RETURN TO CALLER
333 END
```

## Appendix IV

### *AIV.16 Analog APSD Software Modules: NEWTEK Source Code.*

```

1      SUBROUTINE NEW_TK(GRAPHICS)
2      C -----
3      C   THIS MOD IS CALLED "NEWEXP" AND IS ACCESSED BY THE MAIN MENU
4      C   IT WILL SET ALL THE INSTRUMENTATION TO SPECIFIED POSITIONS
5      C   COLLECT AND STORE ALL DATA. FOR EACH SAMPLE THERE WILL BE
6      C   SEVERAL DISCRETE FILES EACH TIED TO ONE ANOTHER.
7      C   1. A HEADER FILE DESCRIBING FACTS ABOUT THE SAMPLE
8      C   2. ALL SAMPLE DATA FOR THE DIFFERENT WAVELENGTHS
9      C   3. AN INDEX FILE SO THAT THE DATA CAN BE REFERENCED.
10     C -----
11
12     CHARACTER TITLE*48,NAME*20,SAMP*20,AGENT*20,CONC*15
13     CHARACTER DATE*9,P1POS*4,P2POS*4,MOD1*2,MOD2*2,ST*3,ERR*4
14     CHARACTER OPT_NUM*2,OPT_INPO*3,OPTIC*1,START_POS*10
15     CHARACTER ARM*8,INCR*8,ENDARM*8,RS,ACK,ANS,LAMDA*15
16     CHARACTER PNAME*3,PLNM*7,SAMPLE,FILENM*20,PORT*10
17     CHARACTER GS,CR,MSG*255,START*6,STOP*6,INC*6,TIME*8
18     CHARACTER AMOUNT*15,ANALYTE*20,SEG*3,E*1
19
20     INTEGER CHANGE,ERRMSG,REPORT,TOT_MEM,FREE_MEM
21     INTEGER RECNUM,TEXT,ISHOTS,IN,KK,DUMMY,TP1,STP,TYPE
22     INTEGER LASER,T1,LASERS,GRAPHICS
23     INTEGER TXT_FLG,RESET
24     REAL ARAY,BRAY,CRAY,DRAY,ERAY,REND,RINCR,IARM
25     REAL START1,DIST
26
27     LOGICAL T
28
29     C -----
30     C   ARRAYS ARAY,BRAY,CRAY,DRAY ARE REDUNDANT DATA ARRAYS THAT ARE
31     C   FOR THE PRESENT USED TO CHECK FOR DISCREPANCIES IS THE DATA.
32     C   THESE ARE THE TAGGED TO THE POLARIZER POSITIONS.
33     C
34     C   VERT/VERT   - ARAY   9 ELEMENTS X 180 degs - 1620 RECORDS
35     C   VERT/45 deg - BRAY
36     C   45 deg/VERT - CRAY
37     C   45 deg/45 deg - DRAY
38     C   ALL DATA   - ERAY  16 ELEMENTS X 180 degs - 2880 RECORDS
39     C -----
40
41     COMMON /MATRIX/ARAY(3600),BRAY(3600),CRAY(3600),DRAY(3600),
42     .ERAY(3600)
43
44     DIMENSION SAMP(10),AGENT(10),CONC(10),LAMDA(40),LAS(40)
45     DIMENSION REAT(16),HEX(16),START(10),STOP(10),INC(10)
46     DIMENSION LASERS(10)
47
48     1  FORMAT('1',/111111T16,A80,/1)
49     2  FORMAT(A20)
50     3  FORMAT(A4)
51     4  FORMAT('0')

```

# Appendix IV

```

52 5  FORMAT(A2)
53 6  FORMAT (A)
54 7  FORMAT(15)
55 8  FORMAT (A10)
56
57 RS - CHAR(30)
58 ACK - CHAR(6)
59 CS - CHAR(29)
60 CR - CHAR(13)
61 E - CHAR(27)
62
63
64
65 TITLE-'Welcome to the Muller Matrix Ellipsometry Experiment'
66
67 CHANGE - 0
68
69 IF(GRAPHICS.EQ.0)GOTO 20
70
71
72 C _____
73 C          ***** TEST DATA *****
74 NAME - 'CHAS'
75 DATE - '7-JAN-90'
76 TIME - '17:30:00'
77 NOSAMP - 1
78 SAMP(1) - 'GOLD'
79 AGENT(1)- 'CB'
80 CONC(1) - '.123 mu'
81 START(1) - '80.'
82 STOP(1) - '90.'
83 INC(1) - '1.0'
84
85 LAS(1) - 1
86 LAS(2) - 2
87 LAS(3) - 3
88 LAS(4) - 4
89
90 LAMDA(1) - '111.1'
91 LAMDA(2) - '222.2'
92 LAMDA(3) - '333.3'
93 LAMDA(4) - '444.4'
94
95 LASERS(1) - 4
96
97 C IF(1123.EQ.0)GOTO 300
98
99 C _____
100 C THIS CHECKS THE TERMINAL TYPE. IF THE USER IS USING A TEKTRONIX
101 C TERMINAL THE ROUTINE WILL USE A COLOR GRAPHICS ROUTINE DESIGNED TO
102 C GET ALL THE SAMPLE INFORMATION FOR UP TO 8 SAMPLES.
103 C _____
104 C
105 C CALL TERM_INPO(ITERM,NUM_PLANES,TOT_MEM,FREE_MEM,

```

## Appendix IV

```

106 C .IVERSION,OPT_NUM,OPT_INPO)
107 C
108 C IF(NUM_PLANES.EQ.-1)THEN
109 C     GOTO 20          ! NOT A TEKTRONIX TERMINAL
110 C
111 C _____
112 C THIS PART CALLS THE GRAPHICS ROUTINE THAT TAKES THE SAMPLE DATA.
113 C _____
114 C
115 C ELSE
116 C
117 C     READ(*,2)DUMMY      ! THIS IS A DUMMY READ
118 C
119 C     CALL TEK_INPUTS(NOSAMP,SAMP,AGENT,CONC,START,STOP,INC,
120 C                     NAME,DATE,TIME,TEXT)
121 C
122 C     GRAPHICS = 1        ! FLAG THAT GRAPHICS ARE
123 C                     ! IN USE.
124 C _____
125 C
126 C     IF(TEXT.EQ.1)GOTO 10000
127 C
128 C _____
129 C THIS IS THE GRAPHICS ROUTINE THAT TAKES SPECIFIC LASER DATA FOR
130 C EACH SAMPLE. HOW MANY LASERS, WHAT ORDER, WHAT WAVELENGTHS
131 C _____
132 C
133 C     CALL LASER_IN(SAMP,AGENT,CONC,NOSAMP,NAME,LASERS,LAMDA,LAS,
134 C                  TEXT)
135 C
136 C     IF(TEXT.EQ.1)GOTO 10000
137 C
138 C     GOTO 500            ! GO BEGIN EXPERIMENT
139 C
140 C
141 C _____
142 C THIS GETS THE NAME OF THE PERSON RUNNING THE EXPERIMENT
143 C _____
144 C
145 20 PRINT 1,TITLE
146 30 FORMAT(T3,' ENTER NAME          : ',5)
147 40 WRITE(*,30)
148 READ(*,2) NAME          ! INPUT NAME
149 WRITE(*,4)
150 IF (CHANGE.EQ.1) GOTO 300
151
152 C _____
153 C THIS GETS THE DATE THE DATA WAS TAKEN
154 C _____
155 C
156 42 FORMAT(T3,' ENTER DATE          : ',5)
157 43 WRITE(*,42)
158 READ(*,2) DATE          ! INPUT DATE
159 WRITE(*,4)

```

# Appendix IV

```

160      IF (CHANGE.EQ.1) GOTO 500
161
162      C-----
163      C   THIS GETS THE NUMBER OF SAMPLES AND THE NAME OF EACH SAMPLE
164      C-----
165
166      50  FORMAT(TS,' ENTER NUMBER OF SAMPLES   : ',S)
167      60  WRITE(*,50)
168      READ (*,ERR=60)NOSAMP      ! INPUT NUMBER OF SAMPLES
169      WRITE(*,4)
170
171
172      WRITE(*,*)'IF ALL SAMPLES ARE THE SAME TYPE < Y > '
173      READ(*,6)ANS
174      IF (ANS.EQ.'Y') GOTO 110
175
176
177      DO 90 I=1,NOSAMP
178
179      70  FORMAT(TS,' ENTER SAMPLE NAME',I,'       : ',S)
180      80  WRITE(*,70)
181      READ(*,2)SAMP(I)          ! INPUT SAMPLE NAME
182      WRITE(*,4)
183      90  CONTINUE
184
185      IF (CHANGE.EQ.1) GOTO 500
186      GOTO 130
187
188      100  FORMAT(TS,' ENTER SAMPLE NAME           : ',S)
189      110  WRITE(*,100)
190      READ (*,2)SAMP(1)          ! INPUT SAMPLE NAME
191      DO 120 I=2,NOSAMP
192      SAMP(I)-SAMP(1)
193      120  CONTINUE
194      WRITE(*,4)
195      IF (CHANGE.EQ.1) GOTO 500
196
197      C-----
198      C   THIS GETS THE TYPE OF AGENT THAT WILL BE USED ON THE SAMPLES
199      C-----
200
201      DO 125 I = 1,NOSAMP
202      AGENT(I) = ''
203      125  CONTINUE
204
205      WRITE(*,*)
206
207      130  WRITE(*,*)'WILL AGENT TYPE BE THE SAME FOR ALL SAMPLES
208      1 < Y > '
209      READ (*,4,ERR=130)ANS
210      IF (ANS.EQ.'Y') GOTO 180
211
212      DO 140 I=1,NOSAMP
213      140  FORMAT(TS,' ENTER AGENT',I,'           : ',S)

```

# Appendix IV

```

214 150 WRITE(*,140)I
215 READ(*,2)AGENT(I) ! INPUT AGENT NAME
216 WRITE(*,4)
217 160 CONTINUE
218
219 IF (CHANGE.EQ.1) GOTO 500
220 GOTO 200
221
222
223 170 FORMAT(T3,' ENTER AGENT NAME : ',S)
224 180 WRITE(*,170)
225
226 READ(*,2) AGENT(I) ! INPUT AGENT NAME
227
228 DO 190, I=2,NOSAMP
229 AGENT(I)-AGENT(1)
230 190 CONTINUE
231
232 WRITE(*,4)
233
234 IF (CHANGE.EQ.1) GOTO 500
235
236 C-----
237 C THIS GETS THE AGENT CONCENTRATION OF THE AGENT FOR ALL THE SAMPLES
238 C-----
239
240 200 WRITE(*,*)'WILL AGENT CONCENTRATION BE THE SAME FOR ALL SAMPLES'
241 WRITE(*,*) < Y >
242 READ(*,6,ERR=200)ANS
243
244 IF(ANS.EQ.'Y') GOTO 250
245
246
247 210 FORMAT(T3,' ENTER CONC. OF AGENT(Mg/m3) : ',S)
248 220 DO 230 I=1,NOSAMP
249 WRITE(*,210)I
250 READ(*,8)CONC(I)
251 WRITE(*,4)
252 230 CONTINUE
253 WRITE(*,4)
254 IF (CHANGE.EQ.1) GOTO 500
255 GOTO 280
256
257 240 FORMAT(T3,' ENTER CONC. OF AGENT(Mg/m3) : ',S)
258 250 WRITE(*,240)
259 READ(*,8)CONC(1)
260
261 DO 260 I=2,NOSAMP
262 CONC(I)-CONC(1)
263 260 CONTINUE
264
265 WRITE(*,4)
266
267 IF (CHANGE.EQ.1) GOTO 500

```

# Appendix IV

```

268
269 C-----
270 C   THIS PART GETS THE START,STOP AND INCREMENT POSITION OF THE SAMPLE
271 C   FOR THE ROTATION IN FRONT OF THE LASER.
272 C-----
273
274 270  FORMAT(10X,'START POSITION OF ARM <DEGREES>: ',S)
275
276      DO 325 I = 1,NOSAMP      ! LOOP THRU SAMPLES
277
278          WRITE(*,332)
279 280      WRITE(*,270)
280
281          READ*,'(A6),ERR-280)START(I) ! START POSITION OF GONIOMETER
282          WRITE(*,4)
283
284 290  FORMAT(10X,'INCREMENT OF ARM <DEGREES> : ',S)
285
286 300      WRITE(*,290)
287          READ*,'(A6),ERR-300)INC(I) ! INCREMENT OF GONIOMETER
288          WRITE(*,4)
289
290 310  FORMAT(10X,'END POSITION OF ARM(DEGREES) : ',S)
291
292 320      WRITE(*,310)
293          READ*,'(A6),ERR-320)STOP(I) ! END POSITION OF GONIOMETER
294          WRITE(*,4)
295
296      IF (CHANGE.EQ.1) GOTO 500
297
298 325  CONTINUE
299
300 C-----
301 C   THIS PART GETS THE LASER WAVELENGTH AND NUMBER. THE NUMBER
302 C   IS ONE OF THE FOUR LASERS BEING USED FOR THIS PARTICULAR EXP.
303 C-----
304
305 330  FORMAT(10X,'ENTER NUMBER OF LASER TO BE USED: ',S)
306 332  FORMAT(10X,'SAMPLE No. ',I2,' SAMPLE: ',A20)
307
308      LCNT = 0      ! THIS IS AN ARRAY INCREMENTER
309
310      DO 380 I = 1,NOSAMP      ! LOOP THRU THE SAMPLES
311
312 340      WRITE (*,332),SAMP(I)
313          WRITE (*,330)
314
315          READ*,'ERR-340)LASERS(I) ! HOW MANY LASERS
316          WRITE(*,4)
317
318 C-----
319 C   THIS PART LOOPS THRU THE LASERS JUST DEFINED AND GETS THE
320 C   ACTUAL LASER NUMBER 1-4 AND ITS WAVELENGTH.
321 C-----

```

# Appendix IV

```

322
323 343 FORMAT(5X,'ENTER THE WAVELENGTH FOR THIS LASER: ',5)
324 346 FORMAT(5X,'ENTER THE LASER NUMBER (1 - 4) : ',5)
325
326 DO 370 J = 1,LASERS(I) ! LOOP THRU THE LASERS
327
328 343 WRITE (C,346)I
329 READ(C,*,ERR=343)LAS(I + LCNT) ! GET LASER NUMBER
330 WRITE(C,4)
331
332 344 WRITE (C,345)
333 READ(C,*,ERR=343)LAMDA(I + LCNT) ! GET LASER WAVELENGTH
334 WRITE(C,4)
335
336 370 CONTINUE ! END LASER LOOP
337
338 LCNT = LCNT + 4 ! INCREMENT ARRAY COUNTER
339
340 380 CONTINUE
341
342 C-----
343
344 IF (CHANGE.EQ.1) GOTO 500
345
346 C-----
347 C THIS PART LETS THE USER SEE IF HE WANTS TO MAKE CORRECTIONS
348 C AND PROVIDES A MEANS OF DOING SO.
349 C-----
350
351 400 FORMAT (24/)
352 410 FORMAT (T34,'HEADER FILE',/)
353 420 FORMAT (10X,'1. OPERATOR: ',A20,10X,'2. DATE: ',A9,' ',A8,/)
354 430 FORMAT (T9,'3. SAMPLE',T36,'4. AGENT',T60,'5. CONC',/)
355 440 FORMAT (3X,I2,3X,A20,8X,A20,10X,A15)
356 450 FORMAT (30X,I2,' LASERS')
357 460 FORMAT (10X,'LASER # ',I2,10X,A15)
358 470 FORMAT (T30,'- CONIOMETER - < DEG >')
359 480 FORMAT (T11,'7. START: ',T22,A6,T31,'8. STOP: ',T41,A6,
360 1 T49,'9. INCREMENT: ',T64,A6,/)
361
362
363 C-----
364 C THIS SHOWS THE USER THE INPUTS THAT WERE JUST MADE ON THE SAMPLES
365 C HERE EACH SAMPLE IS PICTURED INDIVIDUALLY WITH ITS DATA.
366 C-----
367
368 LCNT = 0
369 STRT = 1
370
371 500 DO 550 I=STRT,NOSAMP ! LOOP THRU EACH SAMPLE
372
373 WRITE(C,400)
374 WRITE(C,410)
375 WRITE(C,4)

```

# Appendix IV

```

376      WRITE(*,420)NAME,DATE,TIME
377      WRITE(*,430)
378      WRITE(*,440)I,SAMP(I),AGENT(I),CONC(I)
379      WRITE(*,4)
380      WRITE(*,450)LASERS(I)
381
382      DO 510 J = 1,LASERS(I)
383          WRITE(*,460)LAS(I + LCNT),LAMDA(I + LCNT)
384 510    CONTINUE
385
386      WRITE(*,4)
387      WRITE(*,470)
388      WRITE(*,480)START(I),STOP(I),INC(I)
389      WRITE(*,4)
390
391      C-----
392
393
394 560    FORMAT(10X,'ANY CHANGES ? ( RETURN FOR NONE
395      . OR SELECT NUMBER ) ',5)
396
397 570    WRITE(*,560)
398
399      READ(*,'(A)',ERR=570)ERR
400
401      IF(ERR.EQ.'')GOTO 550
402
403      READ(ERR,'(BN,I2)')ERRMSG
404
405      IF (ERRMSG.GT.9)GOTO 570
406      IF (ERRMSG.GT.0) THEN
407          CHANGE = 1
408          ISTRT = I
409          GOTO (40,45,60,130,200,340,280,320,300) ERRMSG
410      ENDIF
411      CHANGE = 0
412 340    LCNT = LCNT + 1
413 350    CONTINUE
414
415      C-----
416      C   AT THIS POINT CONTROL DATA IS SENT TO ADJUST THE INSTRUMENTS
417      C
418      C-----
419      C   1. OPEN THE SERIAL PORT THRU WHICH ALL THE INSTRUMENTATION I/O
420      C   AND DATA WILL BE CONTROLLED AND COLLECTED.
421      C-----
422
423 572    FORMAT(////,10X,'THE COMMUNICATIONS PORT MUST BE DEPIED WITH')
424 573    FORMAT(10X,'THE FOLLOWING PARAMETERS: ',//)
425 574    FORMAT(30X,'9600 BAUD      NO PARITY')
426 575    FORMAT(30X,'8  BITS      1 STOP BIT',//)
427 576    FORMAT(10X,'THE PORT MUST BE IN A " PASSALL " MODE ',//)
428 578    FORMAT(10X,'PRESS RETURN FOR DEFAULT PORT < TXA2 > ',//)
429 579    FORMAT(10X,'ENTER THE SERIAL PORT NAME: ',5)

```

# Appendix IV

```

430
431 C-----
432 C IF THE USER HAS A TEK TERMINAL THEN THE TEK_TEXT ROUTINE IS
433 C CALLED TO DISPLAY THE ABOVE TEXT.
434 C-----
435
436 580 IF(GRAPHICS.EQ.1)THEN
437     TXT_PLG - 1           ! FLAG FOR PORT SET PARAMETERS
438     PORT - ''           ! CLEAR OUT ANY PORT DATA
439     RESET - 0           ! FLAG TO DRAW A RED PANEL
440     IEXIT - 0           ! INITIALIZE EXIT FLAG
441
442     CALL TEK_TEXT(TXT_PLG,PORT,RESET,ANS,ANALYTE,AMOUNT,IEXT)
443
444     IF(IEXT.EQ.1)GOTO 10000 ! USER WANTS TO EXIT
445
446     GOTO 588
447
448     ENDIF
449 C-----
450 C THIS TEXT IS WRITTEN TO A NON TEKTRONIX TERMINAL.
451 C-----
452
453 583 WRITE(*,572)
454     WRITE(*,573)
455     WRITE(*,574)
456     WRITE(*,575)
457     WRITE(*,576)
458     WRITE(*,578)
459     WRITE(*,579)
460     READ(*, '(A10),ERR=583)PORT
461
462     IF(PORT.EQ.' ')PORT - 'TXA2'
463
464 588 OPEN(3,FILE = PORT,STATUS='NEW',CARRIAGECONTROL = 'NONE',
465     .ERR = 10000)
466
467 C-----
468 C THIS INITIALIZES THE RELAY BANKS.
469 C-----
470
471     WRITE(3, '(A9)GS//%WC16,0//CR
472     WRITE(3, '(A9)GS//%WC70,0//CR
473
474 C-----
475 C INITIALIZE MODULATOR #1. TO KEEP THE MODULATORS CALM WE
476 C DISCOVERED THAT PRIOR TO TURNING ONE OFF ITS GOOD PRACTICE TO
477 C TURN ANOTHER ON FIRST. HERE I GET THE FIRST ONE WARMED UP.
478 C-----
479
480     WRITE(3, '(A9)GS//%WC20,1//CR ! MODULATOR #1 ON
481
482 C-----
483 C 2. THIS PART REQUESTS THAT THE USER DECIDE THE METHOD BY WHICH

```

# Appendix IV

```

484 C      THE EXPERIMENT WILL BE SEEN. I HAVE PROVIDED A METHOD
485 C      BY WHICH REAL TIME GRAPHICS MAY BE GENERATED IF THE
486 C      EXPERIMENT IS CONDUCTED ON A TEKTRONIX TERMINAL
487 C      MODELS 4111 OR ABOVE. ADDITIONALLY, THERE IS A METHOD
488 C      FOR VIEWING THE A/D VOLTAGES FOR EACH DATA CHANNEL PER
489 C      DATA COLLECTION CYCLE. THE USER MAY OPT FOR A "QUIET"
490 C      CYCLE WHERE NO DATA IS PRESENTED TO SPEED UP THE DATA
491 C      COLLECTION IF THE RUN TIME CONFIDENCE IS HIGH.
492 C_____
493
494 581  FORMAT(///,5X,'ENTER THE TYPE OF OUTPUT DESIRED:',//)
495
496 582  FORMAT(10X,'1. REAL TIME A/D CHANNEL VOLTAGE OUTPUTS',//)
497 584  FORMAT(10X,'2. NO DISPLAY OF DATA',//)
498 586  FORMAT(10X,'3. EXIT ROUTINE',//)
499
500 C_____
501 C      THIS PART CALLS THE GRAPHICS ROUTINE TO DISPLAY THE TEXT ABOVE
502 C      AND RETURNS ITS ANSWER IN "ANS".
503 C_____
504
505      IF(GRAPHICS.EQ.1)THEN
506          TXT_FLG - 2          ! FLAG FOR DISPLAY PARAMETERS
507          RESET  - 1          ! FLAG NOT TO DRAW THE PANEL
508          EXITT  - 0          ! INITIALIZE EXIT FLAG
509
510          CALL TEK_TEXT(TXT_FLG,PORT,RESET,ANS,ANALYTE,AMOUNT,EXITT)
511
512          IF(EXITT.EQ.1)GOTO 10000 ! USER WANTS TO EXIT
513          GOTO 589
514
515      ENDIF
516
517 C_____
518 C      THIS IS THE NON TEKTRONIX TEXT TO GET THE TYPE OF OUTPUT THE USER
519 C      IS REQUESTING
520 C_____
521
522 993  WRITE(*,581)
523      WRITE(*,582)
524      WRITE(*,584)
525      WRITE(*,586)
526
527      READ*,'(A),ERR - 993)ANS
528
529      IF(ANS.EQ.'1'.OR.ANS.EQ.'2'.OR.ANS.EQ.'3')GOTO 589
530
531      GOTO 993
532
533 C_____
534
535
536 989  IF(GRAPHICS.EQ.1)THEN      ! USER HAS A TEK TERMINAL
537

```

## Appendix IV

```

538      IF(ANS.EQ.'1')THEN      ! USER WANTS REAL TIME GRAPHICS
539      STP - 1                  ! SET THE DISPLAY FLAG FOR TEK
540
541      ELSEIF(ANS.EQ.'2')THEN    ! USER WANTS A/D VOLTAGES
542      GRAPHICS - 0             ! FLAG THAT NO GRAPHICS REQUESTED
543      STP - 2                  ! SET DISPLAY FLAG FOR CHART MODE
544      RESET - 2                ! DELETE THE RED PANEL
545      CALL TEK_TEXT(TXT_PLG,PORT,RESET,ANS,
546                  ANALYTE,AMOUNT,IEXT)
547
548      ELSEIF(ANS.EQ.'3')THEN    ! USER DOESNT WANT ANY OUTPUT
549      STP - 3                  ! SET THE FLAG FOR QUIET MODE
550      RESET - 2                ! DELETE THE RED PANEL
551      GRAPHICS - 0             ! FLAG THAT NO GRAPHICS REQUESTED
552
553      CALL TEK_TEXT(TXT_PLG,PORT,RESET,ANS,
554                  ANALYTE,AMOUNT,IEXT)
555
556      ENDIF
557
558      C _____
559      C   THE USER DOES NOT HAVE A GRAPHICS TERMINAL
560      C _____
561
562      ELSE
563
564      IF(ANS.EQ.'1')THEN ! USER WANTS A/D VOLTAGES
565      STP - 2           ! SET DISPLAY FLAG FOR CHART MODE
566
567      ELSEIF(ANS.EQ.'2')THEN ! USER DOESNT WANT ANY OUTPUT
568      STP - 3           ! SET THE FLAG FOR QUIET MODE
569
570      ELSEIF(ANS.EQ.'3')THEN ! USER WANTS TO EXIT
571      GOTO 10000        ! RETURN TO CALLER
572
573      ENDIF
574
575      ENDIF
576
577      C _____
578      C   3. THIS PART ESTABLISHES CONTACT WITH THE A/D CONVERTER VIA
579      C   THE UPLINK CONTROLLER CARD #4. THIS IS DONE BY SENDING A
580      C   COMMAND < CHAR(29) // D4 >. THIS CHANNELS ALL SERIAL DATA
581      C   TO THE A/D BOARD.
582      C   THE A/D IS AWAKENED BY 2 CARRIAGE RETURNS UPON WHICH IT
583      C   RETURNS A HYPHEN. THE COMMAND < ST-701 EXECUTIVE ON >
584      C   IS ISSUED BY THE HOST COMPUTER TO ENTER EXECUTIVE MODE
585      C   AND A STAR * * * PROMT IS ISSUED BY THE A/D, IF ALL IS
586      C   WELL. AT THIS POINT THE A/D IS READY TO COLLECT DATA.
587      C   IF THERE IS A PROBLEM THE REPORT FLAG WILL - 1.
588      C _____
589
590      990  RCNT - 0              ! INITIALIZE THE REPEAT COUNTER
591      REPORT - 0                ! INITIALIZE THE REPORT FLAG

```

# Appendix IV

```

592      T1  - 1
593
594 C   WRITE(*,*)T1,REPORT,PORT,IC1,CHAN,RDAT,HEX
595      PORT - 'TXA2'
596
597 C   PAUSE CALLING DATEL TO INITIALIZE IT
598
599
600      CALL DATEL(T1,REPORT,PORT,IC1,CHAN,RDAT,HEX)
601      ! GO AWAKE THE A/D CONVERTER
602
603      IF(REPORT.EQ.1)THEN      ! THERE IS TROUBLE
604      IF(ICNT.EQ.3)GOTO 10000    ! IF OVER 3 TRIES ..QUIT
605      WRITE(*,*)
606      WRITE(*,*) 'THERE IS A PROBLEM WAKING THE A/D CONVERTER'
607      WRITE(*,*)
608      WRITE(*,*) 'CHECK THE CONNECTION AND PRESS RETURN '
609      WRITE(*,*)
610      WRITE(*,*)
611
612      READ*,'(A)',ERR=585)ANS    ! READ THE INPUT
613
614 585      ICNT - ICNT + 1      ! COUNT THE TRIES
615      GOTO 590                ! TRY THE A/D AGAIN
616      ENDIF
617
618 C-----
619 C   4. THIS BEGINS THE LASER CALIBRATION SEQUENCE. HERE A PREDEFINED
620 C       FILE " SPECTRUM.DAT " CONTAINS THE STAGE POSITIONS
621 C       NECESSARY TO CHANNEL THE LASER BEAM FROM EACH LASER INTO
622 C       A SPECTRUM ANALYZER. IF THIS FILE DOES NOT EXIST THEN
623 C       THE USER IS INSTRUCTED TO GO TO THE CALIBRATION ROUTINE
624 C       DEFINED IN THE MAIN MENU.
625 C-----
626
627      REPORT - 0              ! INITIALIZE ERROR FLAG
628
629      WRITE(3, '(A4)')GS//D2//CR      ! INITIALIZE THE OPTIC
630 C                                  ! STAGE CONTROLLER CARD
631      WRITE(3, '(A2)')E '            ! SEND STARTUP COMMAND
632      WRITE(3, '(A2)')E '            ! SEND STARTUP COMMAND
633
634      WRITE(3, '(A9)')E MN ST0 '      ! TURN ON CONTROLLER
635      ! PLACE IN NORMAL MODE
636 C-----
637 C   THIS PART MAKES THE USER CALIBRATE THE LASERS THAT WILL BE USED
638 C   ON THIS SAMPLE. THIS IS DONE ONCE FOR EACH SAMPLE UNLESS A
639 C   CALIBRATION IS NOT NEEDED. THAT IS IF SAMPLE 2+ ON LASER #1 IS
640 C   THE SAME AS LASER #1 ON SAMPLE 1'
641 C-----
642
643      DO 995 I = 1,LASERS(1)      ! LOOP THRU LASERS
644
645      OPTIC - CHAR(48 + LAS(I + ICNT)) ! ACTUAL LASER NUMBER

```

# Appendix IV

```

646      LASER = LAS(I + LCNT)      ! HOLD IN BUFFER
647      TYPE = 1                  ! FLAG FOR SPECTRUM SETUP
648
649
650      CALL MOV_STAGE(LASER,TYPE,LAMDA(I),START_POS,REPORT,
651                    GRAPHICS)
652
653 995  CONTINUE
654
655
656  C-----
657  C   THIS ERASES THE RED TEXT SCREEN
658  C-----
659
660      RESET = 2                  ! DELETE THE RED PANEL
661      CALL TEK_TEXT(TXT_FLG,PORT,RESET,ANS,
662                    ANALYTE,AMOUNT,TEXT)
663
664  C-----
665  C
666  C   INITIALIZATION IS COMPLETE AND THE INSTRUMENTS ARE READY TO
667  C   COLLECT DATA.
668  C   THIS IS THE BEGINNING OF THE MAIN DATA COLLECTING LOOP
669  C-----
670
671      LCNT = 0                    ! LASER ARRAY INCREMENTER
672
673      DO 1070 LOOP = 1,NOSAMP     ! LOOP THRU ALL SAMPLES
674
675  C-----
676  C   THIS CONVERTS THE CHARACTER INPUTS FOR START, END AND INCREMENT
677  C   ANGLES TO REAL NUMBERS.
678  C-----
679
680 600  READ(STOP(LOOP),'(BN,P6.2)')REND
681      READ(START(LOOP),'(BN,P6.2)')RARM
682      READ(INC(LOOP),'(BN,P6.2)')RINCR
683
684      ISHOTS = IFIX((REND - RARM)/RINCR) + 1 ! THIS IS THE TOTAL No
685
686  C-----
687  C   THIS CALLS THE GRAPHICS ROUTINE TO DRAW A BAR GRAPH. THE USER
688  C   HAS THE OPTION SELECT ALL OR JUST SPECIFIC MATRIX ELEMENTS
689  C   TO BE GRAPHED FOR THE EXPERIMENT.
690  C-----
691
692      IF(LOOP.EQ.1.AND.I.CT.1)TP1 = 1
693
694      IF(STP.EQ.1)THEN
695
696          CALL TEK(TP1,SAMP(LOOP),AGENT(LOOP),CONC(LOOP),LAMDA(I))
697
698      ENDIF
699

```

# Appendix IV

```

700 C-----
701 C THIS PART ESTABLISHES IF THERE IS GOING TO BE ANY CHEMISTRY
702 C APPLIED TO THE SAMPLE. IF SO THE SAMPLE STAGE MUST BE MOVED TO
703 C THE APPARATUS THAT SUPPLIES THE CHEMICALS.
704 C-----
705
706 IF(AGENT(LOOP).EQ.'NONE')THEN
707     RESET - 2             ! FLAG TO CLEAR THE SCREEN
708     CALL TEK_TEXT(TXT_PLG,PORT,RESET,ANS,ANALYTE,AMOUNT,TEXT)
709     GOTO 605             ! NO AGENT FOR THIS SAMP
710 ENDIF
711
712 C-----
713 C IF THERE IS GOING TO BE AGENT APPLIED THEN WE COME HERE
714 C I TURN OFF THE OPTIC STAGE CONTROLLER. TURN ON THE SAMPLE STAGE
715 C CONTROLLER AND MOVE THE THE SAMPLE 180000 STEPS COUNTER CLOCKWISE.
716 C THIS EQUALS A FULL 90 DEGREE ROTATION.
717 C-----
718
719 WRITE(3, '(A2)')P '      ! DISABLE OPTIC CONTROLLER
720 CALL TWAIT(3)
721 WRITE(3, '(A4)')GS//DO//CR      ! INITIALIZE CONTROLLER
722 CALL TWAIT(3)
723 WRITE(3, '(A4)')GS//D3//CR      ! INITIALIZE THE SAMPLE
724                                ! CONTROLLER CARD
725 CALL TWAIT(3)
726 WRITE(3, '(A9)')E MN ST0 '      ! TURN ON CONTROLLER
727                                ! PLACE IN NORMAL MODE
728                                ! AND POWER DOWN MOTORS.
729 CALL TWAIT(3)
730
731 IDIST - 180000
732 CALL INT_TO_CHAR(IDIST,START_POS,LE) ! CONVERT IT TO CHARACTER
733
734 TYPE - 3             ! FLAG TO MOVE SAMPLE
735                     ! STAGE ONLY
736 CALL MOV_STAGE(LASER,TYPE,LAMDA(1),START_POS,REPORT,
737               GRAPHICS)
738
739 C-----
740 C SEND A MESSAGE TO THE USER TO APPLY THE CHEMISTRY AT THIS POINT.
741 C THIS IS PRESENTLY DONE MANUALLY. JUST WAIT FOR A RETURN TO BE
742 C ENTERED.
743 C-----
744
745 620 FORMAT(10X,'THE SAMPLE IS NOW READY FOR THE ',A20)
746 622 FORMAT(10X,'APPLY ',A10,' (Mg/m3) TO THE SAMPLE'//)
747 624 FORMAT(10X,'PRESS TO GO ON:')
748
749 C-----
750 C IF THE USER IS USING GRAPHICS THEN THE ABOVE TEXT IS DISPLAYED
751 C IN GRAPHIC FORM BY CALL THE TEK TEXT ROUTINE
752 C-----
753

```

# Appendix IV

```

734 626 IF(GRAPHICS.EQ.1)THEN
735
736 C   PAUSE' ABOUT TO MOVE THE SAMPLE'
737
738 IF(LOOP.EQ.1)THEN
739     RESET - 0           ! FLAG TO DRAW A RED PANEL
740 ELSE
741     RESET - 1           ! FLAG NOT TO DRAW A RED PANEL
742 ENDIF
743
744 IEXT - 0                ! INITIALIZE EXIT FLAG
745
746 ANALYTE - AGENT(LOOP)
747 AMOUNT - CONC(LOOP)
748
749 TXT_FLG - 3             ! FLAG FOR SAMPLE PARAMETERS
750
751 CALL TEK_TEXT(TXT_FLG,PORT,RESET,ANS,ANALYTE,AMOUNT,IEXT)
752
753 IF(IEXT.EQ.1)GOTO 10000 ! USER WANTS TO EXIT
754
755 C _____
756 C   THIS CLEARS THE ENTIRE SCREEN PRIOR TO GOING ON WITH THE PROGRAM
757 C _____
758
759 RESET - 2               ! FLAG TO CLEAR THE SCREEN
760
761 CALL TEK_TEXT(TXT_FLG,PORT,RESET,ANS,ANALYTE,AMOUNT,IEXT)
762
763 GOTO 628
764
765 ENDIF
766
767 WRITE(*,620)AGENT(LOOP) ! TELL THE USER TO ADD
768 WRITE(*,622)CONC(LOOP)  ! THE SPECIFIED CHEMISTRY
769 WRITE(*,624)
770 MSG(1:1) - ''           ! CLEAR THE VARIABLE
771
772
773 CALL TWAIT(50)
774 C   READ*,(A),ERR=628)MSG ! READ THE MESSAGE
775
776 IF(MSG(1:1).EQ.'')GOTO 628 ! GO SEND STAGE HOME
777
778 GOTO 626                 ! INPUT BAD ...DO IT AGAIN
779
780 C _____
781 C   THIS MOVES THE SAMPLE STAGE BACK TO ITS HOME POSITION.
782 C   IT ADDITIONALLY DISABLES THE SAMPLE STAGE CONTROLLER.
783 C   THEN WAKES THE OPTIC CONTROLLER CARD AND PLACES THE OPTIC
784 C   STAGE CONTROLLER IN NORMAL MODE.
785 C _____
786
787 628 IDIST - -180000

```

# Appendix IV

```

808      CALL INT_TO_CHAR(IDIST,START_POS,LE) ! CONVERT IT TO CHARACTER
809
810      TYPE = 3
811      CALL MOV_STAGE(LASER,TYPE,LAMDA(I),START_POS,REPORT,
812                    GRAPHICS)
813
814      CALL TWAIT(0)
815      WRITE(3, '(A2)') 'P' ! DISABLE SAMPLE CONTROL
816      CALL TWAIT(0)
817
818      605 WRITE(3, '(A4)') 'CS1/DO1/ICR' ! INITIALIZE CONTROLLER
819      CALL TWAIT(0)
820      WRITE(3, '(A4)') 'CS1/DO2/ICR' ! INITIALIZE THE OPTIC
821      C ! STAGE CONTROLLER CARD
822      CALL TWAIT(0)
823      WRITE(3, '(A9)') 'E MN ST0' ! TURN ON CONTROLLER
824      CALL TWAIT(0) ! ALL MOTORS DEENERGIZED
825
826
827      C _____
828
829      L = LOOP ! SHORT SAMPLE COUNTER
830      K = 1 ! INITIALIZE LASER COUNTER
831
832      C _____
833      C K = COUNT ON THE NUMBER OF LASERS USED IN THE EXPERIMENT
834      C HERE THE OPTIC POSITIONS ARE DETERMINED BY THE < LAS(K) >
835      C _____
836
837      610 K = K + LCNT ! INCREMENT LASER COUNTER
838
839      LASER = LAS(K)
840
841      C _____
842      C THIS SETS THE TRANSLATION STAGES IN THE PROPER POSITIONS TO
843      C SEND THE BEAM TO THE SAMPLE.
844      C _____
845
846      615 TYPE = 2 ! FLAG TO LINE BEAM WITH SAMPLE
847
848      CALL MOV_STAGE(LASER,TYPE,LAMDA(I),START_POS,REPORT,
849                    GRAPHICS)
850
851      IF(REPORT.GE.1)GOTO 10000 ! EXIT PROGRAM ON ERROR
852
853      C _____
854      C HERE THE SAMPLE STAGE IS SENT TO ITS START POSITION.
855      C _____
856
857      START1 = 90.0 - RARM ! DEGREES TO START POSITION
858
859      IDIST = IPD( START1 * 2000.0 ) ! STEPPER MOTOR INCREMENTS
860      IHOME = IDIST
861

```

# Appendix IV

```

862      CALL INT_TO_CHAR(DIST,START_POS,LE) ! CONVERT IT TO CHARACTER
863
864      WRITE(3, '(A2)')P
865
866      CALL TWAIT(3)
867      WRITE(3, '(A4)')GS//DO//CR      ! INITIALIZE CONTROLLER
868      CALL TWAIT(3)
869      WRITE(3, '(A4)')GS//DS//CR      ! INITIALIZE THE SAMPLE
870      ! CONTROLLER CARD
871      CALL TWAIT(3)
872
873      WRITE(3, '(A9)')E MIN STO      ! TURN ON CONTROLLER
874      ! PLACE IN NORMAL MODE
875      ! AND POWER DOWN MOTORS.
876      CALL TWAIT(3)
877
878      TYPE - 3
879      CALL MOV_STAGE(LASER,TYPE,LAMDA(I),START_POS,REPORT,
880      GRAPHICS)
881
882      C-----
883      C THIS PART OPENS THE PROPER SHUTTER SO THAT THE PORPER LASER BEAM
884      C CAN CHANNEL ITS WAY TO THE SAMPLE. BITS 16 - 19 REPRESENT
885      C CONTACT CLOSURES FOR LASER SHUTTERS 1 - 4 RESPECTIVELY.
886      C-----
887
888      CALL TWAIT(2)
889
890      IF(LASER.EQ.1)THEN
891
892          WRITE(3, '(A9)')GS//%WC20,1//CR      ! MODULATOR #1 ON
893          CALL TWAIT(2)
894          WRITE(3, '(A9)')GS//%WC21,0//CR      ! MODULATOR #2 OFF
895          CALL TWAIT(2)
896          WRITE(3, '(A9)')GS//%WC22,0//CR      ! MODULATOR #3 OFF
897          CALL TWAIT(2)
898          WRITE(3, '(A9)')GS//%WC23,0//CR      ! MODULATOR #4 OFF
899
900
901          WRITE(3, '(A9)')GS//%WC16,1//CR      ! LASER SHUTTER #1 ON
902          CALL TWAIT(2)
903          WRITE(3, '(A9)')GS//%WC17,0//CR      ! LASER SHUTTER #2 OFF
904          CALL TWAIT(2)
905          WRITE(3, '(A9)')GS//%WC18,0//CR      ! LASER SHUTTER #3 OFF
906          CALL TWAIT(2)
907          WRITE(3, '(A9)')GS//%WC19,0//CR      ! LASER SHUTTER #4 OFF
908
909
910      ELSEIF(LASER.EQ.2)THEN
911
912
913          WRITE(3, '(A9)')GS//%WC21,1//CR      ! MODULATOR #2 ON
914          CALL TWAIT(2)
915          WRITE(3, '(A9)')GS//%WC20,0//CR      ! MODULATOR #1 OFF

```

# Appendix IV

```

916      CALL TWAIT(2)
917      WRITEQ,'(A97)GS/1%WC22,0//CR      ! MODULATOR #3 OFF
918      CALL TWAIT(2)
919      WRITEQ,'(A97)GS/1%WC23,0//CR      ! MODULATOR #4 OFF
920      CALL TWAIT(2)
921
922      WRITEQ,'(A97)GS/1%WC16,0//CR      ! LASER SHUTTER #1 OFF
923      CALL TWAIT(2)
924      WRITEQ,'(A97)GS/1%WC17,1//CR      ! LASER SHUTTER #2 ON
925      CALL TWAIT(2)
926      WRITEQ,'(A97)GS/1%WC18,0//CR      ! LASER SHUTTER #3 OFF
927      CALL TWAIT(2)
928      WRITEQ,'(A97)GS/1%WC19,0//CR      ! LASER SHUTTER #4 OFF
929
930      ELSEIF(LASER.EQ.3)THEN
931
932      CALL TWAIT(2)
933      WRITEQ,'(A97)GS/1%WC22,1//CR      ! MODULATOR #3 ON
934
935      WRITEQ,'(A97)GS/1%WC20,0//CR      ! MODULATOR #1 OFF
936      CALL TWAIT(2)
937      WRITEQ,'(A97)GS/1%WC21,0//CR      ! MODULATOR #2 OFF
938      CALL TWAIT(2)
939      WRITEQ,'(A97)GS/1%WC23,0//CR      ! MODULATOR #4 OFF
940      CALL TWAIT(2)
941      WRITEQ,'(A97)GS/1%WC16,0//CR      ! LASER SHUTTER #1 OFF
942      CALL TWAIT(2)
943      WRITEQ,'(A97)GS/1%WC17,0//CR      ! LASER SHUTTER #2 OFF
944      CALL TWAIT(2)
945      WRITEQ,'(A97)GS/1%WC18,1//CR      ! LASER SHUTTER #3 ON
946      CALL TWAIT(2)
947      WRITEQ,'(A97)GS/1%WC19,0//CR      ! LASER SHUTTER #4 OFF
948
949
950
951      ELSEIF(LASER.EQ.4)THEN
952
953      WRITEQ,'(A97)GS/1%WC23,1//CR      ! MODULATOR #4 ON
954      CALL TWAIT(2)
955      WRITEQ,'(A97)GS/1%WC20,0//CR      ! MODULATOR #1 OFF
956      CALL TWAIT(2)
957      WRITEQ,'(A97)GS/1%WC21,0//CR      ! MODULATOR #2 OFF
958      CALL TWAIT(2)
959      WRITEQ,'(A97)GS/1%WC22,0//CR      ! MODULATOR #3 OFF
960      CALL TWAIT(2)
961      WRITEQ,'(A97)GS/1%WC16,0//CR      ! LASER SHUTTER #1 OFF
962      CALL TWAIT(2)
963      WRITEQ,'(A97)GS/1%WC17,0//CR      ! LASER SHUTTER #2 OFF
964      CALL TWAIT(2)
965      WRITEQ,'(A97)GS/1%WC18,0//CR      ! LASER SHUTTER #3 OFF
966      CALL TWAIT(2)
967      WRITEQ,'(A97)GS/1%WC19,1//CR      ! LASER SHUTTER #4 ON
968
969      ENDIF

```

# Appendix IV

```

970
971 C-----
972 C REESTABLISH THE CONNECTION WITH THE #3 SERIAL NODE CONTROLLER
973 C-----
974
975 CALL TWAIT(3)
976 WRITE(3, '(A4)GS//D3//CR' )
977 CALL TWAIT(2)
978
979
980 C-----
981 C HERE THE DATA COLLECTION BEGINS. ALWAYS IN THE SAME ORDER
982 C-----
983
984 INUSE = 0
985
986 CALL VV (ISHOTS, INUSE, ICNT, IS, SAMP, RINCR, RARM, REND,
987 .ANGLE, STP, PORT)
988
989 C CALL PRINT_JT
990
991 CALL V4S (ISHOTS, INUSE, ICNT, IS, SAMP, RINCR, RARM, REND,
992 .ANGLE, STP)
993 CALL P4SV (ISHOTS, INUSE, ICNT, IS, SAMP, RINCR, RARM, REND,
994 .ANGLE, STP)
995 CALL P4AS (ISHOTS, INUSE, ICNT, IS, SAMP, RINCR, RARM, REND,
996 .ANGLE, STP)
997
998
999
1000 C-----
1001 C THIS PART DELETES THE CHANNEL GRAPHICS THAT WERE CREATED ON THE
1002 C LAST POLARIZER PERMUTATION.
1003 C-----
1004
1005 IF (GRAPHICS.EQ.1) THEN
1006 ISEC = 830 ! SEGMENT NUMBER
1007 CALL INTRPT (ISEC, SEG) ! CONVERT TO TEK CODE
1008 WRITE(, 'E//SK//SEG' ) ! DELETE THE SEGMENT
1009 WRITE(, 'E//IGN' ) ! RENEW THE VIEW
1010 ENDIF
1011 C-----
1012 C THIS PART CLOSES THE LASER SHUTTER THAT IS CURRENTLY OPEN
1013 C-----
1014
1015 CALL TWAIT(2)
1016 IF (LASER.EQ.1) THEN
1017
1018 WRITE(, '(A9)GS//WC16,0//CR' ) ! LASER SHUTTER #1 OFF
1019
1020 ELSEIF (LASER.EQ.2) THEN
1021 WRITE(, '(A9)GS//WC17,0//CR' ) ! LASER SHUTTER #2 OFF
1022
1023 ELSEIF (LASER.EQ.3) THEN

```

# Appendix IV

```

1024      WRITE(3, '(A9)') GS//%WC18,0//CR      ! LASER SHUTTER #3 OFF
1025
1026
1027      ELSEIF(LASER.EQ.4) THEN
1028
1029      WRITE(3, '(A9)') GS//%WC19,0//CR      ! LASER SHUTTER #4 OFF
1030
1031      CALL TWAIT(2)
1032      WRITE(3, '(A9)') GS//%WC20,1//CR      ! MODULATOR #1 ON
1033
1034      CALL TWAIT(2)
1035
1036      WRITE(3, '(A9)') GS//%WC23,0//CR      ! MODULATOR #4 OFF
1037
1038      ENDP
1039      CALL TWAIT(2)
1040
1041      C-----
1042      C   REESTABLISH THE CONNECTION WITH THE #3 SERIAL NODE CONTROLLER
1043      C-----
1044
1045      CALL TWAIT(3)
1046      WRITE(3, '(A4)') GS//D3//CR      !
1047      CALL TWAIT(2)
1048
1049      C-----
1050      C   THIS PART SENDS ALL THE STAGES TO THEIR HOME POSITION SO THAT THE
1051      C   NEXT LASER CAN START FRESH.
1052      C-----
1053
1054      WRITE(3, '(A7)') '1ST1 C '
1055
1056      IDIST = 10000
1057      MSG = ' '
1058      IHOME = IHOME - ( 2 * IHOME )
1059
1060      CALL INT_TO_CHAR(IHOME, START_POS, LE) ! CONVERT IT TO CHARACTER
1061
1062      TYPE = 3
1063      CALL MOV_STAGE(LASER, TYPE, LAMDA(I), START_POS, REPORT,
1064                    GRAPHICS)
1065
1066      WRITE(3, '(A21)') '2PS 2ST1 2H 2G 2CR C '
1067      CALL TWAIT(3)
1068
1069      READ(3, '(A80)', ERR=630) MSG
1070
1071      630 CALL TWAIT(3)
1072      MSG = ' '
1073
1074      WRITE(3, '(A5)') '2ST0 '
1075      CALL TWAIT(3)
1076
1077      WRITE(3, '(A21)') '3PS 3ST1 3H 3G 3CR C '

```

# Appendix IV

```

1078     CALL TWAIT(3)
1079     READ(3, '(A80)', ERR=640) MSG
1080
1081 640   WRITE(3, '(A5)') 35T0 '
1082     MSG - ' '
1083     CALL TWAIT(3)
1084
1085     WRITE(3, '(A2)') 'F '      ! TURN OFF THE SAMPLE CONTROLLER
1086     CALL TWAIT(3)
1087
1088
1089     IF (GRAPHICS.EQ.1) THEN
1090         E = CHAR(27)
1091         ISEG = 850
1092         CALL INTRPT(ISEG, SEG)
1093         WRITE(*, *) E // 'SK' // ISEG
1094         WRITE(*, *) E // 'IGN0'
1095     ENDIF
1096
1097 C-----
1098 C   THIS PART KEEPS A TOTAL COUNT OF THE DATA RECORDS ON FILE
1099 C-----
1100
1101 900   INQUIRE(FILE='INDEX', EXIST=T) ! IS THERE A FILE HEADER NUMBER
1102
1103     IF (.NOT.T) THEN
1104         RECNUM = 2      ! THE FILE HAS NOT YET GOT DATA
1105
1106         OPEN (UNIT=1, FILE='INDEX', ACCESS='DIRECT', STATUS='UNKNOWN',
1107             .FORM='FORMATTED', RECL=134)
1108
1109         WRITE(1, REC=1, PMT=7, ERR=970) RECNUM ! TOTAL RECORD NUMBER
1110
1111         WRITE(*, *) 'THE CURRENT RECORDS ON FILE ARE ', RECNUM - 1
1112         GOTO 940
1113     ENDIF
1114
1115 C-----
1116
1117
1118     OPEN (1, FILE='INDEX', ACCESS='DIRECT', STATUS='UNKNOWN',
1119         .FORM='FORMATTED', RECL=134)
1120
1121     READ(1, REC=1, PMT=7, ERR=920) RECNUM
1122
1123 920   WRITE(*, *) 'THE CURRENT RECORDS ON FILE ARE ', RECNUM
1124
1125 C-----
1126 C   BELOW THE INDEX FILE IS WRITTEN TO AND WILL CONTAIN
1127 C   THE DATE, SAMPLE, AGENT, AND THE LASER WAVE LENGTHS
1128 C   CONCENTRATION VALUES WILL BE LISTED WITH HEADER DATA
1129 C-----
1130
1131 936   FORMAT(B, A20, A9, A8, 2(A20), A15, A15, 4, 3(P6.2))

```

# Appendix IV

```

1132
1133 940 WRITE(1,REC-RECNUM,PMT-956,ERR-970)RECNUM,NAME,DATE,TIME,
1134 .SAMP(L),AGENT(L),CONC(L),LAMDA(K+LCNT),ISHOTS,RARM,REND,RINCR
1135
1136 WRITE(1,REC-1,PMT-7,ERR-970)RECNUM+1 ! INC & WRITE COUNT TO FILE
1137 970 CLOSE (1)
1138
1139 C-----
1140 C
1141 C NOW THE ACTUAL DATA IS STORED. IT WILL SAVE AS MANY FILES FOR
1142 C EACH SAMPLE AS THERE ARE LASERS (DIFFERENT WAVELENGTHS). EACH
1143 C TIME THIS MOD IS REACHED ONE SAMPLE HAS BEEN LOOKED AT.
1144 C
1145 C-----
1146
1147 995 L1 = 0
1148
1149 CALL INT_TO_CHAR(RECNUM,PNAME,LE)
1150 FILENM=PNAME/'DAT'
1151
1152 DO 1000 J=1,20
1153 IF (FILENM(J:J).EQ.CHAR(32))GOTO 1000
1154 L1=L1+1
1155 FLNM(L1:L1)-FILENM(J:J) ! SHORTEN THE FILE NAME
1156 1000 CONTINUE
1157
1158 FILENM = ''
1159 PNAME = ''
1160
1161 1005 FORMAT(F7.5)
1162
1163 OPEN(1,FILE=FLNM,ACCESS='DIRECT',RECL=7,FORM='FORMATTED',
1164 . STATUS='NEW',ERR=1090)
1165
1166 N1 = ISHOTS * 10
1167
1168 1010 DO 1015 KK = 1,N1
1169 WRITE(1,REC-KK,PMT-1005,ERR-1015)ARRAY(KK)
1170 1015 CONTINUE
1171
1172 J = 0
1173 I2 = KK
1174
1175 1020 DO 1025 KK = I2,N1 + I2 - 1
1176 J = J + 1
1177 WRITE(1,REC-KK,PMT-1005,ERR-1025)BRAY(J)
1178 1025 CONTINUE
1179
1180 J = 0
1181 I2 = KK
1182 1030 DO 1035 KK = I2,N1 + I2 - 1
1183 J = J + 1
1184 WRITE(1,REC-KK,PMT-1005,ERR-1035)CRAY(J)
1185 1035 CONTINUE

```

# Appendix IV

```

1186
1187     J - 0
1188     I2 - KK
1189
1190 1040 DO 1045 KK - I2, I1 + I2 - 1
1191     J - J + 1
1192     WRITE(1, REC-KK, PMT-1005, ERR-1045) DRAY()
1193 1045 CONTINUE
1194
1195     I1 - ISHOTS * 16
1196     I2 - KK
1197     J - 0
1198
1199 1050 DO 1055 KK - I2, I1 + I2 - 1
1200     J - J + 1
1201     WRITE(1, REC-KK, PMT-1005, ERR-1055) ERAY()
1202 1055 CONTINUE
1203
1204     CLOSE(1)
1205
1206 C-----
1207 C   NOW I SWITCH CONTROL TO THE OPTIC STAGE CONTROLLER
1208 C   IF THIS IS LASER #1 NO ACTION NEED BE DONE.
1209 C   FOR LASERS 2 - 4 I SEND THE STEPPER MOTOR #1 OPTIC STAGE HOME
1210 C   IF THIS IS LASER #3 I SEND STAGE #2 -63000 STEPS HOME
1211 C   IF THIS IS LASER #4 I SEND STAGE #2 -126000 STEPS HOME..
1212 C-----
1213
1214     WRITE(3, '(A4)'CS//DO'//CR      ! INITIALIZE CONTROLLER
1215     CALL TWAIT(3)
1216     WRITE(3, '(A4)'CS//D2'//CR      ! INITIALIZE THE OPTIC
1217                                     ! STAGE CONTROLLER CARD
1218     CALL TWAIT(3)                   ! PAUSE 3/10 SEC
1219     WRITE(3, '(A9)'E MN ST0 '      ! TURN ON CONTROLLER
1220     CALL TWAIT(3)
1221
1222 C-----
1223 C   THIS IS THE END OF THE LASER LOOP. WHATEVER LASER WAS LAST TO
1224 C   TAKE DATA MUST BE SENT HOME BEFORE WE MOVE TO THE NEXT SAMPLE.
1225 C-----
1226
1227     IF (K.GE.LASERS(LOOP))THEN      ! THIS WAS THE LAST LASER
1228     IF(LAS(K).EQ.1)GOTO 1090        ! IF LASER #1 DO NOTHING
1229     GOTO 1060                       ! ELSE MOVE ON..
1230     ENDP
1231
1232 C-----
1233 C   IF THIS IS LASER #1 AND NOT THE LAST LASER THEN WE INCREMENT THE
1234 C   LASER COUNT AND GO GET THE NEXT LASER DATA. NO MOTORS HAVE
1235 C   TO BE MOVED.
1236 C-----
1237
1238     IF(LAS(K).EQ.1)THEN
1239     K - K + 1

```

# Appendix IV

```

1240      GOTO 610          ! DO NOTHING IF LASER #1
1241      ENDIF
1242
1243      C----- HERE I SEND STEPPER #1 TO ITS HOME -----
1244
1245      1060  WRITE(*,*)'LASER - ',LAS(K)
1246      WRITE(*, '(A35)')'1P5 15T1 1V20 1A35 1D52000 G 1X1 C '
1247      CALL TWAIT(3)      ! PAUSE 3/10 SEC
1248      READ(*, '(A235)')MSG
1249      WRITE(*,*) MSG - ',MSG
1250      CALL TWAIT(3)
1251      WRITE(*, '(A5)')'15T0 '
1252      CALL TWAIT(3)      ! PAUSE 3/10 SEC
1253
1254      C-----
1255      C   IF LASER IS NUMBER 3 I SEND IT HOME
1256      C-----
1257
1258      IF(LAS(K).EQ.3)THEN
1259      WRITE(*, '(A36)')'2P5 25T1 2V20 2A35 2D-63000 G 2X1 C '
1260      WRITE(*,*) AT A READ SENDING MOTOR #2 HOME'
1261      CALL TWAIT(3)
1262      READ(*, '(A80)')MSG
1263      WRITE(*,*) MSG - ',MSG
1264
1265      WRITE(*, '(A5)')'25T0 '
1266      CALL TWAIT(3)
1267      ENDIF
1268
1269      C-----
1270      C   IF LASER IS NUMBER 4 I SEND IT HOME HERE
1271      C-----
1272
1273      IF(LAS(K).EQ.4)THEN
1274      WRITE(*, '(A37)')'2P5 25T1 2V20 2A35 2D-126000 G 2X1 C '
1275      WRITE(*,*) AT A READ SENDING MOTOR #2 HOME'
1276      CALL TWAIT(3)
1277      READ(*, '(A80)')MSG
1278
1279      WRITE(*,*) MSG - ',MSG
1280
1281      WRITE(*, '(A5)')'25T0 '
1282      CALL TWAIT(3)
1283      ENDIF
1284
1285      C-----
1286
1287      IF (K.GE.LASERS(LOOP))THEN ! THIS WAS THE LAST LASER
1288      LCNT = LCNT + 4          ! INCREMENT THE ARRAY COUNTER
1289      GOTO 1090              ! GOTO NEXT SAMPLE IF ANY
1290      ENDIF
1291      C-----
1292      C   HERE WE CONTINUE THRU THE LASER LOOP WHERE WE ASK THE NEXT
1293      C   LASER TO SHINE ON THE SAMPLE

```

# Appendix IV

```

1294 C_____
1295
1296 K - K + 1 ! INCREMENT LASER COUNTER
1297 GOTO 610 ! GOTO NEXT LASER IF THERE ARE ANY
1298
1299
1300 C_____
1301 C THIS MAKES THE USER TUNE THE LASER FOR THE NEXT SAMPLE.
1302 C_____
1303
1304 1090 DO 1100 IJ - 1,LASERS(I + 1) ! LOOP THRU LASERS
1305
1306 OPTIC - CHAR(48 + LAS(IJ + LCNT)) ! ACTUAL LASER NUMBER
1307 LASER - LAS(IJ + LCNT) ! HOLD IN BUFFER
1308 TYPE - 1 ! FLAG FOR SPECTRUM SETUP
1309
1310
1311 CALL MOV_STAGE(LASER,TYPE,LAMDA(IJ),START_POS,REPORT,
1312 GRAPHICS)
1313
1314 1100 CONTINUE
1315
1316
1317 C_____
1318 C THIS ERASES THE RED TEXT SCREEN
1319 C_____
1320
1321 RESET - 2 ! DELETE THE RED PANEL
1322 CALL TEK_TEXT(TXT_PLG,PORT,RESET,ANS,
1323 ANALYTE,AMOUNT,TEXT)
1324
1325 C_____
1326 C THIS IS THE BOTTOM OF THE SAMPLE LOOP
1327 C_____
1328
1329 1070 CONTINUE
1330
1331 WRITE(3, '(A2)YF' ! DISABLE CONTROLLER
1332 WRITE(3, '(A4)CS//DO//CR ! INITIALIZE CONTROLLER
1333 ! COMMUNICATIONS
1334
1335 C_____
1336
1337 IFLAG-0
1338
1339
1340 10000 CLOSE(3) ! CLOSE CONTROLLER COM PORT
1341
1342 E - CHAR(27)
1343 WRITE(*, 'E//RPO' ! SET FIXUP LEVEL - 0
1344 WRITE(*, 'E//LV0' ! DISABLE THE DIALOG AREA
1345 WRITE(*, 'E//SKY' ! DELETE ALL SEGMENTS
1346 WRITE(*, 'E//RPM' ! RESET THE FIXUP LEVEL
1347 WRITE(*, 'E//KNO' ! RENEW THE VIEW

```

## Appendix IV

1348 10010 RETURN ! RETURN TO CALLER  
1349  
1350 END

## Appendix IV

### *AIV.17 Analog APSD Software Modules: P4545 Source Code.*

```

1      SUBROUTINE P4545(ISHOTS,INUSE,ICNT,IS,SAMP,RINCR,RARM,REND,
2      .ANGLE,STP)
3
4      C-----
5      C   THIS MOD CAPTURES THE DATA FOR THE POLARIZERS IN POSITION
6      C   45 deg/45 deg. TWO ARRAYS ARE USED TO SAVE THE DATA.
7      C   DRAY - IS A SEQUENTIAL ARRAY STORING ALL THE DATA RECEIVED FROM
8      C   THE A/D CONVERTER THRU THE ENTIRE ROTATION OF THE SAMPLE.
9      C
10     C   ERAY - IS A REPRESENTATION OF THE ENTIRE MATRIX. NINE ELEMENTS
11     C   WILL BE PASSED IN FROM THE A/D CONVERTER OF WHICH ONLY 1 WILL
12     C   BE NEW. IT IS: ( 11 ) FROM THE 1,3,4,9,11,12,13,15,16.
13     C   THIS WILL COMPLETE THE ERAY DATA COLLECTION.
14     C-----
15
16
17     CHARACTER GETDAT*4,SAMP*20,MSG*50,MESC*255,HEX*4
18     CHARACTER PORT*10,CHAN*2
19
20     REAL RDAT,ARAY,BRAY,CRAY,DRAY,ERAY,RINCR,REND,ANGLE,RARM
21
22     INTEGER STP,TYPE,REPORT
23
24     DIMENSION GETDAT(16),RDAT(16),HEX(16)
25
26     COMMON /MATRIX/ARAY(3600),BRAY(3600),CRAY(3600),DRAY(3600),
27     . ERAY(5800)
28
29     ISWEEP - 4          ! FLAG THAT THIS IS THE 4TH
30                       ! SWEEP OF THE POLARIZERS
31     C-----
32     C   THIS IS THE LAST POLARIZER SETTING AND DATA COLLECTION ROUTINE
33     C   THE RECEIVER POLARIZER IS SENT TO -45 DEGS. THE SAMPLE STAGE
34     C   IS SET IN THE REVERSE DIRECTION TO TRAVEL BACK TO THE START
35     C   POSITION.
36     C-----
37
38     WRITE(3, '(A4)')'1X0 '      ! RESET THE CUMM POSITION CNTR
39     CALL TWAIT(1)              ! MOVE RECEIVER POLARIZER -45 DEGS
40     WRITE(3, '(A38)')'3F5 35T1 3V10 3A10 3D-225000 3G 3CR C '
41     READ(3, '(A50)')MSG(1:50)  ! WAIT FOR CARRIAGE RETURN
42     CALL TWAIT(1)
43     WRITE(3, '(A5)')'35T0 '      ! DEENERGIZE AXIS 3 MOTOR
44     CALL TWAIT(1)
45     WRITE(3, '(A12)')'1F5 15T1 1H ' ! ENERGIZE AND REVERSE SAMPLE
46     CALL TWAIT(1)
47
48     INUSE1 - 0
49     C-----
50     C   HERE BEGINS THE LOOP WHERE SAMPLE DATA IS TAKEN AND STORED
51     C-----

```

# Appendix IV

```

52
53      DO 200 M - 1,ISHOTS      ! LOOP THRU THE SAMPLE ROTATION
54
55      C-----
56      C   HERE IS WHERE THE A/D CONVERTER IS ASKED FOR THE DATA
57      C-----
58
59      C   CALL TESTDAT(ISWEEP,ANGLE,RDAT)
60
61      IC1 - 11
62      TYPE - 3
63      CALL DATEL(TYPE,REPORT,PORT,IC1,CHAN,RDAT,HEX)
64
65      C-----
66      C   HERE EACH ARRAY IS SELECTED, WRITTEN TO AND INCREMENTED
67      C-----
68
69      DO 100 I - 10,1,-1
70          DRAY(IS) - RDAT(I)      ! STORE THE 10 ARRAY ELEMENTS
71          IS - IS - 1      ! ARAY SEQUENTIAL COUNTER
72 100      CONTINUE
73
74      C-----
75      C   HERE ONLY ONE ELEMENT OF THE ERAY MATRIX IS NEW. IT IS
76      C   ELEMEN" 11. AS STORED IN RDAT IN THE 5TH POSITION.
77      C-----
78
79      K - 11 + ICNT      ! SELECT CORRECT ARRAY ELEMENT
80      ERAY(K) - RDAT(5)      ! PLACE MATRIX DATA IN ERAY
81
82      C-----
83      C   THE STP DETERMINES THE TYPE OF OUTPUT THE USER IS REQUESTING
84      C-----
85
86      IF(STP.EQ.1)THEN      ! USER WANTS REAL TIME GRAPHICS
87          CALL DRAW_ELE(ISWEEP,ANGLE,RARM,RINCR,REND,RDAT,
88              INUSE1)
89          GOTO 150      ! FLAG TO SKIP THE VIEW
90      ENDIF
91
92      IF(STP.EQ.3)GOTO 150      ! NO OUTPUT IS REQUESTED
93
94      CALL VIEW(SAMP,ANGLE,IS,STP,DIR) ! A/D VOLTAGE OUTPUTS
95
96 150      IF(M.EQ.ISHOTS)GOTO 200
97
98      ANGLE - ANGLE - RINCR      ! NEW SAMPLE ANGLE
99
100      ICNT - ICNT - 16      ! DECREMENT THE ARRAY BY 16
101
102      WRITE(3, '(A9)') '1G 1X1 C'      ! MOVE THE SAMPLE
103      READ(3, '(A30)',ERR - 200)MMSG(1:30)
104      CALL TWAFT(1)
105      READ(MMSG(14:22), '(B1,9)',ERR - 200)MOTION

```

## Appendix IV

```

106      ICNG - MOTION - MBUP
107      MBUP - MOTION
108 160  FORMAT(5X,14,10X,' ACCUMULATED MOTION: ',9,
109      ' RELATIVE MOTION: ',9//)
110 D    WRITE(*,160)M,MOTION,ICNG
111      WRITE(3,'(A4)')'1P5 '      ! PAUSE THE STEPPER MOTOR
112
113 200  CONTINUE                  ! LOOP THRU ROTATIONS
114
115 C _____
116 C   AT THIS POINT THE DATA COLLECTION FOR POLARIZERS POSITIONED AT
117 C   45 deg/45 deg IS COMPLETE. WE NOW RETURN TO THE CALLER
118 C _____
119
120      WRITE(3,'(A5)')'15T0 '      ! DEENERGIZE THE SAMPLE STAGE
121
122 400  RETURN
123      END

```

# Appendix IV

## AIV.18 Analog APSD Software Modules: P45V Source Code.

```

1      SUBROUTINE P45V(ISHOTS,INUSE,ICNT,IS,SAMP,RINCR,RARM,REND,
2      .ANGLE,STP)
3
4      C-----
5      C  THIS MOD CAPTURES THE DATA FOR THE POLARIZERS IN POSITION
6      C  45 deg/VERTICLE. TWO ARRAYS ARE USED TO SAVE THE DATA.
7      C  CRAY - IS A SEQUENTIAL ARRAY STORING ALL THE DATA RECEIVED FROM
8      C  THE A/D CONVERTER THRU THE ENTIRE ROTATION OF THE SAMPLE.
9      C
10     C  ERAY - IS A REPRESENTATION OF THE ENTIRE MATRIX. NINE ELEMENTS
11     C  WILL BE PASSED IN FROM THE A/D CONVERTER OF WHICH ONLY 3 WILL
12     C  BE NEW. THEY ARE: (3,7,15) FROM THE 1,3,4,5,7,8,13,15,16.
13     C  ONLY ONE GAP WILL BE LEFT IN THE ARRAY AT THE 11 POSITION WHICH
14     C  WILL BE FILLED IN BY THE LAST POLARIZER SETTING.
15     C-----
16
17
18     CHARACTER GETDAT*4,SAMP*20,MSG*50,MESC*255,HEX*4,PORT*10
19     CHARACTER CHAN*2
20
21     REAL RDAT,ARAY,BRAY,CRAY,DRAY,ERAY,RARM,RINCR,ANGLE,REND
22
23     INTEGER POS,STP,REPORT,TYPE
24
25     DIMENSION RDAT(16),J1(3),HEX(16)
26
27     COMMON /MATRIX/ARAY(3600),BRAY(3600),CRAY(3600),DRAY(3600),
28     .ERAY(5800)
29
30     DATA J1 /3,7,15/      ! NEW MATRIX ARRAY ELEMENTS
31
32     ISWEEP - 3              ! FLAG THAT THIS IS 3RD SWEEP
33                             ! OF THE POLARIZERS
34     C-----
35     C  THIS ADJUSTS THE INSTRUMENTS:
36     C
37     C  THE CUMM POSITION BUFFER IS INITIALIZED
38     C  THE TXMITER POLARIZER IS SENT TO -45 DEGS.
39     C  THE RECEIVER IS SENT TO HOME OR VERTICAL
40     C  THE SAMPLE STAGE IS SET TO REVERSE THE DIRECTION
41     C-----
42
43     WRITEQ,('A4')Y1X0 '      ! RESET CUMM POSITION COUNTER
44     CALL TWAIT(1)
45     WRITEQ,('A21')Y3PS 35T1 3H 3G 3CR C ' ! SEND RECEIVER HOME
46     READQ,('A30')MSG(1:50)      ! WAIT FOR CARRIAGE CNTRL
47     CALL TWAIT(1)
48     WRITEQ,('A5')Y35T0 '      ! DEENERGIZE AXIS #3
49     CALL TWAIT(1)              ! SEND XTMR -45 DEGS
50     WRITEQ,('A37')Y2PS 25T1 2V10 2A10 2D225000 2G 2CR C '
51     READQ,('A30')MSG(1:50)      ! WAIT FOR CARRIAGE CNTRL

```

# Appendix IV

```

52      CALL TWAIT(1)
53      WRITE(3, '(A5)') '2570 '      ! DEENERGIZE AXIS #2
54      CALL TWAIT(1)
55      WRITE(3, '(A13)') '1PS 1ST 1H '      ! REVERSE SAMPLE STAGE
56
57
58      INUSE1 = 0
59
60      C-----
61      C   HERE BEGINS THE LOOP WHERE SAMPLE DATA IS TAKEN AND STORED
62      C-----
63
64      DO 200 M = 1, ISHOTS      ! LOOP THRU THE SAMPLE ROTATION
65
66      C-----
67      C   HERE IS WHERE THE A/D CONVERTER IS ASKED FOR THE DATA
68      C-----
69
70      C   CALL TESTDAT(ISWEEP, ANGLE, RDAT)
71
72      IC1 = 11
73      TYPE = 3
74      CALL DATEL(TYPE, REPORT, PORT, IC1, CHAN, RDAT, HEX)
75
76      C-----
77      C   HERE EACH ARRAY IS SELECTED, WRITTEN TO AND INCREMENTED
78      C-----
79
80      DO 100 I = 1, 10
81          IS = IS + 1      ! ARRAY SEQUENTIAL COUNTER
82          CRAY(IS) = RDAT(I)      ! STORE THE 10 ARRAY ELEMENTS
83      100 CONTINUE
84
85      C-----
86      C   HERE ONLY THREE ELEMENTS OF THE ERAY MATRIX ARE NEW. THEY ARE
87      C   ELEMENTS 3,7,15. THEY ARE STORED IN THE RDAT ARRAY IN POSITIONS
88      C   2,5,8
89      C-----
90
91      K = J1(1) + ICNT      ! SELECT CORRECT ARRAY ELEMENT
92      ERAY(IQ) = RDAT(2)      ! PLACE MATRIX DATA IN ERAY
93      K = J1(2) + ICNT      ! SELECT CORRECT ARRAY ELEMENT
94      ERAY(IQ) = RDAT(5)      ! PLACE MATRIX DATA IN ERAY
95      K = J1(3) + ICNT      ! SELECT CORRECT ARRAY ELEMENT
96      ERAY(IQ) = RDAT(8)      ! PLACE MATRIX DATA IN ERAY
97
98      C-----
99
100     IF (STP.EQ.1) THEN      ! USER WANTS REAL TIME GRAPHICS
101
102     CALL DRAW_ELE(ISWEEP, ANGLE, RARM, RINCR, REND, RDAT, INUSE1)
103     GOTO 150
104 ENDIF
105

```

# Appendix IV

```

106      IF(STP.EQ.3.)GOTO 150
107
108      CALL VIEW(SAMP,ANGLE,IS,STP,DIR)
109
110 150  IF(M.EQ.ISHOTS)GOTO 200      ! DO NOT ROTATE SAMPLE ON LAST
111
112      ANGLE = ANGLE + RINCR      ! NEW SAMPLE ANGLE
113
114      ICNT=ICNT+16      ! INCREMENT THE ARRAY BY 16
115
116
117  C-----
118
119      WRITE(3, '(A9)')'1G 1X1 C'      ! MOVE THE SAMPLE STAGE
120      READ(3, '(A30)',ERR = 200)MSG(1:50)
121      CALL TWAIT(1)
122      READ(MESG(14:22), '(I1N,19)',ERR = 200)MOTION
123      ICNG = MOTION - MBUP
124      MBUP = MOTION
125 170  FORMAT(5X,14,10X,' ACCUMULATED MOTION: ',19,
126      ' RELATIVE MOTION: ',19/)
127  D  WRITE(,170)M,MOTION,ICNG
128      WRITE(3, '(A4)')'1P5'
129      CALL TWAIT(1)
130 200  CONTINUE      ! LOOP THRU ROTATIONS
131
132  C-----
133  C  AT THIS POINT THE DATA COLLECTION FOR POLARIZERS POSITIONED AT
134  C  45 deg/VERTICLE IS COMPLETE. WE NOW RETURN TO THE CALLER
135  C  WHERE THE NEXT POLARIZER SETTING WILL BE MADE AND THE SAMPLE WILL
136  C  BE ROTATED BACKWARDS.
137  C-----
138
139      WRITE(3, '(A5)')'1S70'      ! DEENERGIZE SAMPLE STAGE
140
141 400  RETURN
142      END

```

## Appendix IV

### *AIV.19 Analog APSD Software Modules: READ\_Q10\_1 Source Code.*

```

1      SUBROUTINE READ_Q10(DEVICE,MESG,LENGTH,TIMEOUT,ISIZE,
2          IOUT,CHANNEL,INUSE)
3
4      C _____
5      C  THIS IS A ROUTINE USED FOR VAX VMS HARDWARE TO ALLOW FOR A QUEUED
6      C  IO READ. IN THIS CASE I AM INTERESTED IN READING A SERIAL PORT
7      C  WITH A TIMEOUT SPECIFIED IN SECONDS ( FROM THE USER ).
8      C
9      C  DEVICE - THE SERIAL PORT THAT IS READ FROM      ( IN )
10     C
11     C  METSTR - THE STRING OF DATA THAT IS READ      ( OUT )
12     C
13     C  LENGTH - THE MAXIMUM LENGTH OF THE STRING THATS READ ( IN )
14     C
15     C  TIMEOUT - THE TIME IN SECONDS TO WAIT AT THE PORT FOR DATA ( IN )
16     C
17     C  ISIZE - THE LENGTH OF THE STRING THAT IS READ  ( OUT )
18     C
19     C  IOUT - FLAG THAT HAS 2 OPTIONS:                ( IN/OUT )
20     C
21     C      1. IOUT IS SET IN THE NORMAL MODE TO FLAG WHETHER
22     C          ANY DATA WAS READ FROM THE PORT PRIOR TO TIMING
23     C          OUT. IOUT - 1 NO DATA READ AND PORT TIMED OUT
24     C          IOUT - 0 DATA WAS READ  ( OUT )
25     C
26     C      2. IOUT IS ALSO USED AS A FLAG FROM THE USER WHEN
27     C          THE READ ROUTINE IS COMPLETE. THIS IS DONE SO THAT
28     C          THE ASSIGNED CHANNEL FOR THE SERIAL PORT CAN BE
29     C          DEASSIGNED. IOUT - 99      ( IN )
30     C _____
31
32     IMPLICIT INTEGER*4 (A-Z)
33     INCLUDE 'MODEPY'      ! EXTERNAL VMS VAX DEFINITIONS
34     INCLUDE 'SSSDEPY'     ! EXTERNAL VMS VAX DEFINITIONS
35
36     CHARACTER*512  METSTR      ! STRING RETURNED TO CALL
37     CHARACTER*255  MESG       ! INPUT STRING
38     CHARACTER*10   DEVICE     ! THIS IS THE PORT
39     INTEGER*2      CHANNEL    ! CHANNEL ASSIGNED BY
40                               ! THE SYSTEM
41     INTEGER*2      ISIZE      ! LENGTH OF INPUT STRING
42     INTEGER*2      IOUT       ! FLAG FOR TIMEOUT
43     INTEGER*4      LENGTH     ! LENGTH OF RECEIVED STR
44     INTEGER*4      TIMEOUT    ! PORT TIMEOUT IN SECS
45     INTEGER*4      MASK(2)    ! QUAD WORD ARRAY THAT
46                               ! SETS PORT TERMINATOR
47                               ! CHARACTER.
48     INTEGER*4      FUNCTION    ! FUNCTION FOR Q10 READ
49
50     C _____
51     C  HERE I CREATE A STRUCTURED BLOCK THAT IS USED FOR THE IOSB.

```

## Appendix IV

```

52 C   THIS INCLUDES THE STATUS TO THE QUEUED OPERATION.
53 C   THIS IS NECESSARY BECAUSE THE QIOW MIGHT RETURN A 1 AS STATUS
54 C   THERE WILL BE 4 DIFFERENT TYPES OF STATUS INFORMATION.
55 C   1. IOSTAT   - RETURNS THE STATUS OF THE READ 1 - GOOD
56 C   2. TERM_OFFSET - CHARACTERS READ AT THE PORT
57 C   3. TERMINATOR - ASCII VALUE OF TERMINATOR OR 1ST CHARACTER OF TERM
58 C   4. TERM_SIZE  - LENGTH OF THE TERMINATOR STRING
59 C _____
60
61     STRUCTURE /IOSTAT_BLK/
62     INTEGER*2   IOSTAT,
63     .           TERM_OFFSET,
64     .           TERMINATOR,
65     .           TERM_SIZE
66     END STRUCTURE
67
68     RECORD /IOSTAT_BLK/ IOSTB
69
70 C _____
71 C THIS STRUCTURE BLOCK IS USED TO ALLOW ANY ASCII CHARACTER UP TO 7 BITS
72 C TO BE USED AS A TERMINATION CHARACTER IN A QIO READ OPERATION.
73 C _____
74
75     STRUCTURE /TERM_BLK/      ! CHAR0_15 IS THE NAME FOR THE
76     INTEGER*2   CHAR0_15,     ! TWO BYTES OF THE STRUCTURE. THIS
77     .           CHAR16_31,     ! STRUCTURE IS 16 CONTIGUOUS BYTES
78     .           CHAR32_47,     ! IN MEMORY WITH EACH BIT IN THE
79     .           CHAR48_63,     ! STRUCTURE CORRESPONDING TO THE
80     .           CHAR64_79,     ! ASCII CHARACTER WITH THAT VALUE.
81     .           CHAR80_95,     ! IF WE WANT TO SET THE CHAR "J" TO
82     .           CHAR96_111,    ! BE A TERMINATION CHAR FOR A QIO
83     .           CHAR112_127    ! READ THEN WE SET THE 93RD BIT IN
84     END STRUCTURE             ! THE STRUCTURE BY :
85     .                           ! TERM_CHAR.CHAR80_95 - 2**13
86     RECORD /TERM_BLK/ TERM_CHAR
87
88 C _____
89
90     STRUCTURE /PARAMETER/
91     INTEGER*2   MASK_SIZE,     ! # BYTES IN CHAR0 - CHAR127 FIELDS
92     .           DUMMY          ! NOT USED
93     INTEGER*4   TERM_LOC      ! ADDRESS OF TERM_CHAR STRUCTURE
94     END STRUCTURE
95
96
97     RECORD /PARAMETER/ MASK_P4
98
99     MASK_P4.TERM_LOC = %LOC(TERM_CHAR)
100    MASK_P4.MASK_SIZE = 16      ! # OF BYTES IN CHAR STRUCTURES
101
102    TERM_CHAR.CHAR32_47 = 2**10 ! STAR IS TERMINATOR
103
104    .                               ! SET - 1 IF TIMED OUT
105

```

## Appendix IV

```

106 C _____
107 C  HERE I CHECK THE FLAG "IOUT" FOR A CHANGE IN MODE.  THAT IS A CALL
108 C  TO DEASSIGN THE CURRENT CHANNEL ASSIGNED TO THE PORT.
109 C  THIS IS DONE USUALLY AT THE END OF THE CALLING ROUTINE.
110 C _____
111
112 IF(IOUT.EQ.99)THEN
113     IOUT - 0          ! INITIALIZE TIMEOUT FLAG
114     STATUS - SYS$DASSGN(%VAL(CHANNEL)) ! SYSTEM ASSIGNED CHANNEL
115     IF(.NOT. STATUS) CALL LIB$STOP (%VAL(STATUS))
116     GOTO 1000         ! RETURN TO CALLER
117 ENDIF
118
119 C _____
120 C  THIS IS THE CHANNEL ASSIGNMENT FUNCTION.  THIS CHANNEL IS ASSIGNED
121 C  TO THE DEVICE THAT WILL BE FUNCTIONING WITH THE QUEUED I/O.
122 C  HERE THE DEVICE IS INPUT BY THE USER AND THE SYSTEM ASSIGNS THAT
123 C  DEVICE A CHANNEL.
124 C _____
125
126
127
128 IF(INUSE.EQ.0)THEN          ! ASSIGN CHANNEL ONLY
129     INUSE - 1              ! ONCE.
130     STATUS - SYS$ASSIGN(DEVICE,      ! USER READ DEVICE
131         CHANNEL,      ! SYSTEM ASSIGNED CHANNEL
132         ,              ! PRIVILEGED ACCESS MODE
133         ,)              ! LOGICAL NAME OF MAILBOX
134
135     IF(.NOT. STATUS) CALL LIB$STOP (%VAL(STATUS))
136
137 ENDIF
138
139 C _____
140 C  THIS IS THE QUEUED INPUT/OUTPUT REQUEST.
141 C _____
142
143 IOUT - 0          ! INITIALIZE TIMEOUT FLAG
144 MSG - ''          ! CLEAR OUT THE MESSAGE STRING
145 METSTR - ''       ! CLEAR OUT THE SEND BACK STRING
146
147
148 ! ASSIGN THE FUNCTION OF THE QIOW
149 FUNCTION - IOS_READVBLK.OR.IOSM_TIMED
150
151 ! TO BE A READ
152 STATUS - SYS$QIOW,      ! QIO COMPLETION EVENT FLAG
153     %VAL(CHANNEL), ! ASSIGN THE CHANNEL FOR QIOW
154     %VAL(FUNCTION), ! TYPE OF QIOW REQUESTED BY USER
155     IOSB,          ! IO STATUS BLOCK TO CHECK IO
156     ,              ! AST ROUTINE TO BE EXECUTED
157     ,              ! AST PARAMETERS TO ABOVE ROUTINES
158     %REP(MSG),      ! P1 - INPUT DATA FROM DEVICE
159     %VAL(LENGTH),   ! P2 - LENGTH OF THE INPUT DATA
160     %VAL(TIMEOUT),  ! P3 - PORT TIMEOUT ON THE READ

```

## Appendix IV

```

160      MASK_P4,      ! P4 - SET TERMINATOR CHARACTER
161      ,              ! P5 - CHARACTER FOR READ PROMPT
162      )              ! P6 - SIZE OF THE PROMPT
163
164
165 C-----
166 C   THIS IS THE DIAGNOSTICS SECTION OF THE ROUTINE.
167 C   IOUT - 1 MEANS THAT THE PORT HAS TIMED OUT ON THE READ.
168 C   ISIZE - THE LENGTH OF THE STRING
169 C   METSTR - THE READ DATA SENT BACK TO THE CALLER
170 C   I INCREMENT THE ISIZE VARIABLE TO INCLUDE THE TERMINATOR CHARACTER
171 C-----
172
173      IF(.NOT. STATUS) CALL LIMSTOP (%VAL(STATUS))
174
175      IF(.NOT. IOSB.IOSTAT) THEN
176
177      IF(IOSB.IOSTAT.EQ.SSB_TIMEOUT) THEN
178          IOUT - 1      ! FLAG THAT PORT TIMED OUT
179          GOTO 1000
180      ENDIF
181      ENDIF
182
183 C-----
184
185      ISIZE - IOSB.TERM_OFFSET
186      ISIZE - ISIZE + 1      ! LET ISIZE INCLUDE THE TERMINATOR
187      METSTR - MESC(1:ISIZE) ! CHARACTER
188
189 1000 RETURN
190      END

```

## Appendix IV

### AIV.20 Analog APSD Software Modules: REAL\_to\_CHAR Source Code.

```

1      SUBROUTINE REAL_TO_CHAR (RNUM,ICNT,CNUM)
2
3      C-----
4      C   THIS IS DESIGNED TO TAKE IN A REAL NUMBER(UNDER 10.0)
5      C   AND CONVERT THEM TO A CHARACTER FOR USE WITH A
6      C   GRAPHICS ROUTINE. I TAKE THEM OUT TO THE THOUSANDS PLACE
7      C
8      C   RNUM - ARRAY OF UP TO 16 REAL NUMBERS ( PASSED IN BY CALLER )
9      C
10     C   ICNT - THE NUMBER OF REALS TO BE CONVERTED ( PASSED IN )
11     C
12     C   CNUM - ARRAY OF THE 6 CHARACTER REPRESENTATION OF THE REALS
13     C           ( PASSED BACK TO THE CALLER )
14     C-----
15
16     REAL RNUM,RBUP
17
18     CHARACTER CNUM*6
19
20     DIMENSION RNUM(16),CNUM(16)
21
22     C-----
23
24     DO 50 J = 1,ICNT           ! LOOP THRU REALS
25
26     RBUP = RNUM(J)
27
28     IF(RBUP.LT.0.0)CNUM(J)(1:1) = '-' ! IF NUMBER IS NEGATIVE
29                                     ! WILL REQUIRE A SIGN
30
31     IF(RBUP.GE.10.0)GOTO 50      ! CANT HANDLE NUMBERS
32                                     ! GREATER THAN 10.0
33
34     RBUP = ABS(RBUP)             ! TAKE ABSOLUTE VALUE
35
36     C-----
37     C   THIS PART CHECKS TO SEE IF THE REAL HAS A WHOLE NUMBER PART TO IT.
38     C   IF SO I MAKE THE NUMBER THE SECOND VALUE AND PLACE A DECIMAL IN
39     C   THE THIRD VALUE. THE FIRST CHARACTER IS RESERVED FOR SIGN (+/-)
40     C   IF THERE IS NO WHOLE NUMBER PORTION I PLACE A ZERO IN PLACE 2
41     C   AND DECIMAL IN PLACE 3.
42     C-----
43
44
45     IF(RBUP.GT.0.0)THEN          ! IF > 1 MAKE IT THE
46                                     ! FIRST CHARACTER.
47     INUM = IPX(RBUP)             ! TAKE THE WHOLE NUMBER
48     CNUM(J)(2:3) = CHAR(INUM + 48)/'. ' ! CONVERT TO CHARACTER
49
50     ELSE                         ! OR ELSE
51

```

# Appendix IV

```

52      CNUM(J)(2:3) = '0.'      ! 1ST CHAR IS ZERO
53      ENDRP
54
55      RBUP = RBUP - FLOAT(INUM)      ! SUBTRACT THE WHOLE NUM
56
57      C-----
58      C   THIS PART DEALS WITH THE NUMBERS LESS THAN 1 NE.
59      C   1 LOOP THRU EACH PLACE VALUE BEGINNING WITH 1/10 MULTIPLY THE
60      C   NUMBER BY 10.0 SO I HAVE A WHOLE NUMBER VALUE 0 - 9 THEN
61      C   CONVERT IT TO INTEGER AND SAVE IT AS A CHARACTER IN THE
62      C   CNUM STRING.
63      C-----
64
65      M = 4      ! CHARACTER PLACE COUNTER
66
67      DO 30 K = 1,3      ! LOOP THRU PLACE VALUES
68
69      RBUP = RBUP * 10.0      ! SHIFT NUMBER ONE PLACE LEFT.
70
71      IF(RBUP.GE.1.0)THEN      ! CHECK FOR DATA IN PLACE
72
73      INUM = IPDX(RBUP)      ! SAVE NUMBER AS INTEGER
74
75      CNUM(J)(M:M) = CHAR(INUM + 48) ! CONVERT IT TO CHARACTER
76
77      RBUP = RBUP - FLOAT(INUM) ! SUBTRACT NUMBER FROM TOTAL
78
79      ELSE      ! OR ELSE
80
81      CNUM(J)(M:M) = '0'      ! PLACE IS A ZERO.
82
83      ENDRP
84
85      M = M + 1      ! INCREMENT PLACE COUNTER
86
87      30 CONTINUE
88
89      C-----
90      C   THIS LOOKS AT THE 10,000th PLACE TO SEE IF 1/1000th SHOULD BE
91      C   INCREMENTED UP OR LEFT ALONE. THIS IS NOT A ROUNDING UP
92      C   PROCEDURE. BECAUSE OF THE WAY COMPUTERS STORE REALS THE LAST
93      C   NUMBER MUST BE EVALUATED TO CORRECT FOR THE ORIGINAL INPUT VALUE.
94      C-----
95
96      M = M - 1      ! PUT PLACE POINTER TO 1/1000
97      RBUP = RBUP * 10.0      ! SHIFT NUMBER ONE PLACE LEFT.
98      INUM = IPDX(RBUP)      ! SAVE NUMBER AS INTEGER
99
100
101      IF(INUM.GE.5)THEN      ! CHECK FOR DATA IN PLACE VALUE
102
103      ! IF ABOVE 4 THEN
104      CNUM(J)(M:M) = CHAR(CHAR(CNUM(J)(M:M)) + 1) ! INCREMENT BY 1
105      ! CHECK THE CHARA
106      IF(CNUM(J)(M:M).EQ.'9')CNUM(J)(M:M) = '0' ! VALUE. IF 9 IS

```

## Appendix IV

```
106          ! INCREMENTED ...KEEP IT 9
107      ENDIF
108
109      C _____
110
111      50  CONTINUE
112
113
114      RETURN
115      END
```

## Appendix IV

### AIV.21 Analog APSD Software Modules: SEE\_ELE Source Code.

```

1      SUBROUTINE SEE_ELE(ISWEEP,MATRIX,INU)
2
3      C-----
4      C   THIS MOD LETS THE USER SELECT THE ELEMENTS OF CHOICE FOR EACH
5      C   SWEEP OF THE SAMPLE STAGE. SINCE 9 BITS OF DATA ARE SENT TO
6      C   THE DRAWING ROUTINE EVERY TIME A SAMPLE IS GATHERED THE SCREEN
7      C   CAN BECOME SOMEWHAT CLUTTERED... HERE THE USER HAS THE OPTION
8      C   TO SELECT SPECIFIC MATRIX ELEMENTS OF CHOICE. THE SELECTION
9      C   OF THE DIFFERENT MATRIX ELEMENTS ARE PROVIDED FOR EACH SWEEP
10     C   OF THE SAMPLE STAGE PRIOR TO THE BEGINNING OF THE EXPERIMENT.
11     C-----
12
13
14     CHARACTER TEXT1*35,TEXT2*35,TEXT3*35
15     CHARACTER E,A*5,ANS*12,SEC*3
16
17     INTEGER X,Y,MATRIX
18
19     DIMENSION LDAT1(10),LDAT2(10),LDAT3(10),LDAT4(10)
20     DIMENSION IA(9),IB(9),IC(9),ID(9)
21
22     DATA IA / 1,2,4,5,6,8,13,14,16/
23     DATA IB / 1,2,4,9,10,12,13,14,16/
24     DATA IC / 1,3,4,5,7,8,13,15,16/
25     DATA ID / 1,3,4,9,11,12,13,15,16/
26
27     COMMON LDAT1,LDAT2,LDAT3,LDAT4
28
29     E = CHAR(27)
30
31     C-----
32
33     WRITE(*,*)E//LV0'      ! DISABLE DIALOG AREA
34     WRITE(*,*)E//LZ'      ! CLEAR THE DIALOG AREA
35
36     IF(INU.EQ.1)THEN      ! INUSE BEING RESET FROM OUTSIDE
37         INUSE = 0
38         INU = 0
39     ENDIF
40     C-----
41     C   THIS PART IS JUST AS ADD ON TO THE ROUTINE. IT IS USED BY
42     C   THE ROUTINE DRAW_ELE TO SHOW THE USER WHICH ELEMENTS THAT
43     C   ARE BEING VIEWED. THE FLAG MATRIX WHEN SET - 1 JUMPS THE
44     C   USER TO LINE 13 CHECKS THE SWEEP OF THE POLARIZERS, TURNS OFF THE
45     C   SEGMENTS THAT ARE NOT IN USE THEN THEN RETURNS TO THE CALLER
46     C-----
47
48     IF(MATRIX.EQ.1)THEN
49         WRITE(*,*)E//SV:1'    ! TURN ON ALL SEGMENTS
50         GOTO 13
51     ENDIF

```

# Appendix IV

```

52
53 C-----
54
55 10 IF(INUSE.EQ.0)THEN      ! IF THIS IS THE FIRST TIME IN
56     ISWEEP - 1            ! SET SWEEP FLAG TO 1 INDICATING
57     INUSE - 1             ! SET INUSE FLAG TO ON POSITION
58     ELSE                  ! A VERTICAL, VERTICAL POSITION
59     ISWEEP - ISWEEP + 1    ! OF THE POLARIZERS.
60     IEXIT - 0             ! INITIALIZE THE EXIT FLAG
61     WRITE(*,"E/VS11"      ! TURN ALL SEGMENTS ON
62     ISEG - 800            ! DEFINE TEXT SEGMENT AT 800
63     CALL INTRPT(ISEG,SEG)  ! CONVERT INTEGER TO TEK CHARACTER
64     WRITE(*,"E/PSK//SEG    ! DELETE THE SEGMENT
65     WRITE(*,"E/TKNO"      ! RENEW THE VIEW
66     ENDIF
67
68     J - 1                 ! INITIALIZE ARRAY COUNTER
69     IEXIT - 0             ! INITIALIZE EXIT FLAG
70     IALL - 0              ! INITIALIZE SELECT ALL FLAG
71
72 C-----
73 C THIS PART MAKES THE MATRIX ELEMENTS THAT WILL BE PROVIDING DATA
74 C FOR A PARTICULAR POLARIZER COMBINATION VISIBLE TO THE USER.
75 C THE FOUR POSSIBLE POSITIONS ARE AS FOLLOWS:
76 C
77 C ISWEEP - 1              ! VERTICAL, VERTICAL
78 C ISWEEP - 2              ! VERTICAL, 45 Degrees
79 C ISWEEP - 3              ! 45 Degrees, VERTICAL
80 C ISWEEP - 4              ! 45 Degrees, 45 Degrees.
81 C-----
82
83 13 IF(ISWEEP.EQ.1)THEN
84
85     J - 0                 ! INITIALIZE COUNTER
86
87     DO 20 I = 1,9         ! LOOP THRU THE ARRAY ELEMENTS
88
89 15     J - J + 1           ! INCREMENT THE COUNTER
90
91     IF(J.EQ.17)GOTO 20    ! KEEP UNDER 17
92
93     IF(IA(I).EQ.J)GOTO 20 ! JUMP OUT ON A MATCH
94
95     ISEG - J + 9          ! CALCULATE THE SEGMENT NUMBER
96
97     CALL INTRPT(ISEG,SEG) ! CONVERT IT TO A TEK CHARACTER
98
99     IF(ISEG.LT.16)THEN    ! IF ONLY ONE TEK CHARACTER
100        WRITE(*,"E/VS//SEG(1:1)/" ! ERASE IT
101    ELSE                  ! ELSE IF 2 CHARACTERS
102        WRITE(*,"E/VS//SEG(1:2)/" ! ERASE IT
103    ENDIF
104
105    GO TO 15              ! GO INCREMENT COUNTER

```

# Appendix IV

```

106 20    CONTINUE          ! END LOOP
107
108      TEXT1 - 'POLARIZERS: VERTICAL,VERTICAL'
109
110  C-----
111  C    ISWEEP - 2    ( VERTICAL, 45 DEG )
112  C-----
113
114      ELSEIF(ISWEEP.EQ.2)THEN
115
116          J - 0            ! INITIALIZE COUNTER
117
118          DO 30 I = 1,9      ! LOOP THRU THE ARRAY ELEMENTS
119
120 25      J - J + 1          ! INCREMENT THE COUNTER
121
122          IF(J.EQ.17)GOTO 30 ! KEEP UNDER 17
123
124          IF(TB(I).EQ.J)GOTO 30 ! JUMP OUT ON A MATCH
125
126          ISEG = J + 9      ! CALCULATE THE SEGMENT NUMBER
127
128          CALL INTRPT(ISEG,SEG) ! CONVERT IT TO A TEK CHARACTER
129
130          IF(ISEG.LT.16)THEN ! IF ONLY ONE TEK CHARACTER
131              WRITE(*,"E/SV",ISEG(1:1)/"0" ! ERASE IT
132          ELSE                ! ELSE IF 2 CHARACTERS
133              WRITE(*,"E/SV",ISEG(1:2)/"0" ! ERASE IT
134          ENDP
135
136          GO TO 25          ! GO INCREMENT COUNTER
137
138 30    CONTINUE
139
140      TEXT1 - 'POLARIZERS: VERTICAL, 45 Degrees'
141
142  C-----
143  C    ISWEEP - 3    ( 45 DEG, VERTICAL )
144  C-----
145
146
147      ELSEIF(ISWEEP.EQ.3)THEN
148
149          J - 0
150
151          DO 40 I = 1,9      ! LOOP THRU THE ARRAY ELEMENTS
152
153 35      J - J + 1          ! INCREMENT THE COUNTER
154
155          IF(J.EQ.17)GOTO 40 ! KEEP UNDER 17
156
157          IF(IC(I).EQ.J)GOTO 40 ! JUMP OUT ON A MATCH
158
159          ISEG = J + 9      ! CALCULATE THE SEGMENT NUMBER

```

# Appendix IV

```

160
161      CALL INTRPT(ISEG,SEG)  ! CONVERT IT TO A TEK CHARACTER
162
163      IF(ISEG.LT.16)THEN    ! IF ONLY ONE TEK CHARACTER
164          WRITE(*,"E/5V//SEG(1:1)/0" ! ERASE IT
165      ELSE                  ! ELSE IF 2 CHARACTERS
166          WRITE(*,"E/5V//SEG(1:2)/0" ! ERASE IT
167      ENDIF
168
169      GO TO 35              ! GO INCREMENT COUNTER
170
171 40      CONTINUE
172
173      TEXT1 = 'POLARIZERS: 45 Degrees, VERTICAL'
174
175      C-----
176      C   ISWEEP - 4   ( 45 DEC, 45 DEC )
177      C-----
178
179
180      ELSEIF(ISWEEP.EQ.4)THEN
181
182          J = 0
183
184          DO 50 I = 1,9      ! LOOP THRU THE ARRAY ELEMENTS
185
186 45          J = J + 1      ! INCREMENT THE COUNTER
187
188          IF(J.EQ.17)GOTO 50 ! KEEP UNDER 17
189
190          IF(ID(I).EQ.J)GOTO 50 ! JUMP OUT ON A MATCH
191
192          ISEG = J + 9      ! CALCULATE THE SEGMENT NUMBER
193
194          CALL INTRPT(ISEG,SEG)  ! CONVERT IT TO A TEK CHARACTER
195
196          IF(ISEG.LT.16)THEN    ! IF ONLY ONE TEK CHARACTER
197              WRITE(*,"E/5V//SEG(1:1)/0" ! ERASE IT
198          ELSE                  ! ELSE IF 2 CHARACTERS
199              WRITE(*,"E/5V//SEG(1:2)/0" ! ERASE IT
200          ENDIF
201
202          GO TO 45              ! GO INCREMENT COUNTER
203
204 50      CONTINUE
205
206      TEXT1 = 'POLARIZERS: 45 Degrees, 45 Degrees'
207
208      ENDIF
209
210
211
212      IF(MATRIX.EQ.1)GOTO 1000 ! RETURN TO CALLER JOB DONE
213

```

# Appendix IV

```

214
215 C-----
216 C  THIS PART ASSEMBLES THE CORRECT TEXT TO SHOW THE USER THE POSSIBLE
217 C  SELECTIONS FOR THE CURRENT ARRAY.
218 C-----
219
220 TEXT2 - '1. SELECT ELEMENTS TO VIEW'
221 TEXT3 - '3. SELECTION COMPLETE'
222
223 C-----
224 C  THIS PART DRAWS THE TEXT TO THE DIALOG AREA WINDOW.
225 C-----
226
227 ISEG - 800          ! BEGIN SEGMENT AT 800
228 CALL INTRPT(ISEG,SEG) ! CONVERT INTEGER TO TEK CHARACTER
229 WRITE(*,"E//SE//SEG" ! BEGIN THE SEGMENT
230 WRITE(*,"E//MT1"      ! TEXT COLOR WHITE
231 X - 120              ! X POSITION OF THE ORIGIN
232 Y - 350              ! Y POSITION OF THE ORIGIN
233 CALL HTY(X,Y,A)      ! CONVERT TO TEK CHARACTER
234 WRITE(*,"E//LP//A"    ! SET THE ORIGIN
235 WRITE(*,"E//LTB7//TEXT1" ! WRITE 1ST LINE OF TEXT
236
237 Y - Y - 200          ! DECREMENT Y BY 100 SCREEN UNITS
238 CALL HTY(X,Y,A)      ! INTEGER TO TEK CHARACTER
239 WRITE(*,"E//LP//A"    ! SET THE ORIGIN
240 WRITE(*,"E//LTB7//TEXT2" ! WRITE THE TEXT.
241
242 Y - Y - 100          ! DECREMENT Y BY 100 SCREEN UNITS
243 CALL HTY(X,Y,A)      ! INTEGER TO TEK CHARACTER
244 WRITE(*,"E//LP//A"    ! SET THE ORIGIN
245 WRITE(*,"E//LTB7//TEXT3" ! WRITE THE TEXT.
246 WRITE(*,"E//SC"      ! CLOSE THE SEGMENT
247
248
249 J - 1              ! INITIALIZE PLACE COUNTER
250
251 C-----
252 C  THIS PART CHECKS TO MAKE SURE THE USER HAS SELECTED A CORRECT
253 C  ARRAY ELEMENT, PLACES THE DATA IN THE CORRECT ARRAY ASSOCIATED
254 C  WITH THE SWEEP NUMBER.
255 C-----
256
257 60 READ(*,"(A12),ERR=60)ANS ! READ THE SELECTION
258
259 IF(ANS(1:1).EQ.'1'.OR.ANS(1:1).EQ.'3')GOTO 65
260 GOTO 60
261
262 C-----
263 C  HERE THE USER HAS ELECTED TO END THE ELEMENT SELECTION.
264 C  IF THIS IS NOT THE 4TH SWEEP THEN WE GO SELECT AGAIN.
265 C  OTHERWISE THIS ROUTINE IS COMPLETE.
266 C-----
267

```

# Appendix IV

```

268 65 IF(ANS(1:1).EQ.'3')THEN      ! END THE SELECTION
269     IF(SWEEP.EQ.4)GOTO 280      ! IF LAST SWEEP THEN EXIT
270     GOTO 10                     ! GO TO NEXT SWEEP
271 ENDIF
272
273 C-----
274 C THIS MOD FILLS THE CORRECT ARRAY ( LDAT ) WITH THE CHOSEN
275 C ELEMENTS. J - THE LDAT# ARRAY PLACEMENT. THE LAST ARRAY ELEMENT
276 C CONTAINS THE NUMBER OF USER ELEMENTS SELECTED.
277 C-----
278
279
280 SEG - ANS(7:9)                  ! THIS IS THE SEGMENT SELECTED
281
282 CALL DECODE(SEG,ISEG)           ! CONVERT IT TO AN INTEGER
283
284 IF(ISEG.EQ.40)THEN              ! IF ITS THE " EXIT " BUTTON
285     EXIT - 1                    ! SET THE EXIT FLAG - 1
286     GOTO 70                     ! PLACE THIS INFO IN THE ARRAY
287 ENDIF                           ! NO ELEMENTS ARE TO BE VIEWED
288
289 IF(ISEG.EQ.41)THEN              ! THIS IS THE " ALL " BUTTON
290     IALL - 1                    ! SET THE SHOW ALL ELEMENTS FLAG
291     GOTO 70
292 ENDIF
293
294 IF(ISEG.LT.10.OR.ISEG.GT.25)GOTO 60 ! CHECK FOR CORRECT SEGMENTS
295
296 70 IPICK - ISEG - 9             ! CONVERT THEM TO ELEMENT NUMBERS
297
298 WRITE(*,"*IPICK -",IPICK
299 WRITE(*,"*ISEG - ",ISEG
300
301 C-----
302 C THIS PART PLACES THE THE SELECTED A/D CHANNEL NUMBERS IN THE
303 C CORRECT LDAT FILE. THE LAST ELEMENT IS THE COUNT OF DATA ELEMENTS
304 C THE USER SELECTED. AS THE USER SELECTS THE ELEMENTS I ERASE THE
305 C GRAPHIC SEGMENT ASSOCIATED WITH IT.
306 C-----
307
308 CALL INTRPT(ISEG,SEG)           ! CONVERT SEGMENT INTEGER TO TEK
309
310 IF(ISEG.EQ.40.OR.ISEG.EQ.41)GOTO 80 ! EXIT,ALL -DO NOT ERASE THEM
311
312 IF(ISEG.LT.16)THEN              ! IF ONLY 1 CHARACTER THEN
313     WRITE(*,"E/SV#SEG(1:1)/0" ! ERASE IT.
314 ELSE                            ! OR ELSE IF IT IS 2 CHARACTERS
315     WRITE(*,"E/SV#SEG(1:2)/0" ! ERASE IT ALSO.
316 ENDIF
317
318 80 IF(SWEEP.EQ.1)THEN           ! IF SWEEP #1
319
320 IF(EXIT.EQ.1)THEN              ! USER WANTED TO EXIT
321     LDAT(10) - 0               ! PLACE A 0 IN NUMBER OF

```

# Appendix IV

```

322      GO TO 10          ! ELEMENTS TO VIEW
323      ENDIF            ! GO TO NEXT SWEEP INPUTS
324
325      IF(IALL.EQ.1)THEN  ! USER WANTS TO SEE ALL 9
326      LDAT1(10) = 9      ! ELEMENTS. PLACE THE NUMBER
327      DO 90 I = 1, 9      ! IN PLACE 10, PUT THE 9
328      LDAT1(I) = I        ! CHANNELS IN ORDER IN THE
329  90  CONTINUE           ! ARRAY.
330      IALL = 0           ! INITIALIZE THE ALL ELEMENTS
331      GOTO 10            ! FLAG AND GO TO THE NEXT
332      ENDIF            ! SWEEP
333
334      DO 100 I = 1, 9      ! LOOP THRU ARRAY ELEMENTS
335      IF(IA(I).EQ.IPICK)LDAT1(I) = I ! PUT A/D CHANNEL NUMBER IN
336  100  CONTINUE           ! THE ARRAY
337
338      LDAT1(10) = J        ! PUT CHANNEL COUNT IN LAST HOLE
339      J = J + 1           ! INCREMENT ARRAY PLACE COUNTER
340      IF(J.EQ.10)GOTO 10   ! CHECK TO SEE IF ITS FULL
341      GOTO 60             ! GO BACK FOR MORE DATA
342
343  C-----
344
345      ELSEIF(SWEEP.EQ.2)THEN
346
347      IF(EXTT.EQ.1)THEN    ! USER WANTED TO EXIT
348      LDAT2(10) = 0        ! PLACE A 0 IN NUMBER OF
349      GO TO 10             ! ELEMENTS TO VIEW
350      ENDIF              ! GO TO NEXT SWEEP INPUTS
351
352      IF(IALL.EQ.1)THEN    ! USER WANTS TO SEE ALL 9
353      LDAT2(10) = 9        ! ELEMENTS. PLACE THE NUMBER
354      DO 110 I = 1, 9      ! IN PLACE 10, PUT THE 9
355      LDAT2(I) = I        ! CHANNELS IN ORDER IN THE
356  110  CONTINUE           ! ARRAY.
357      IALL = 0           ! INITIALIZE THE ALL ELEMENTS
358      GOTO 10            ! FLAG AND GO TO THE NEXT
359      ENDIF            ! SWEEP
360
361      DO 120 I = 1, 9      ! LOOP THRU ARRAY ELEMENTS
362      IF(IA(I).EQ.IPICK)LDAT2(I) = I ! PUT A/D CHANNEL NUMBER IN
363  120  CONTINUE           ! THE ARRAY
364
365      LDAT2(10) = J        ! PUT CHANNEL COUNT IN LAST HOLE
366      J = J + 1           ! INCREMENT ARRAY PLACE COUNTER
367      IF(J.EQ.10)GOTO 10   ! CHECK TO SEE IF ITS FULL
368      GOTO 60             ! GO BACK FOR MORE DATA
369
370  C-----
371
372      ELSEIF(SWEEP.EQ.3)THEN
373
374      IF(EXTT.EQ.1)THEN    ! USER WANTED TO EXIT
375      LDAT3(10) = 0        ! PLACE A 0 IN NUMBER OF

```

# Appendix IV

```

376      GO TO 10      ! ELEMENTS TO VIEW
377  ENDIF      ! GO TO NEXT SWEEP INPUTS
378
379  IF(IALL.EQ.1)THEN      ! USER WANTS TO SEE ALL 9
380    LDAT3(10) = 9      ! ELEMENTS. PLACE THE NUMBER
381    DO 130 I = 1, 9      ! IN PLACE 10, PUT THE 9
382      LDAT3(I) = I      ! CHANNELS IN ORDER IN THE
383 130  CONTINUE      ! ARRAY.
384    IALL = 0      ! INITIALIZE THE ALL ELEMENTS
385    GOTO 10      ! FLAG AND GO TO THE NEXT
386  ENDIF      ! SWEEP
387
388  DO 140 I = 1, 9      ! LOOP THRU ARRAY ELEMENTS
389    IF(IC(I).EQ.IPC(I)LDAT3(I)) - I ! PUT A/D CHANNEL NUMBER IN
390 140  CONTINUE      ! THE ARRAY
391
392    LDAT3(10) = J
393    J = J + 1
394    IF(J.EQ.10)GOTO 10
395
396    GOTO 60
397
398  C -----
399
400  ELSEIF(ISWEEP.EQ.4)THEN
401
402  IF(EXT.EQ.1)THEN      ! USER WANTED TO EXIT
403    LDAT4(10) = 0      ! PLACE A 0 IN NUMBER OF
404    GO TO 200      ! ELEMENTS TO VIEW
405  ENDIF      ! GO TO NEXT SWEEP INPUTS
406
407
408  IF(IALL.EQ.1)THEN      ! USER WANTS TO SEE ALL 9
409    LDAT4(10) = 9      ! ELEMENTS. PLACE THE NUMBER
410    DO 150 I = 1, 9      ! IN PLACE 10, PUT THE 9
411      LDAT4(I) = I      ! CHANNELS IN ORDER IN THE
412 150  CONTINUE      ! ARRAY.
413    IALL = 0      ! INITIALIZE THE ALL ELEMENTS
414    GOTO 200      ! FLAG AND GO TO THE NEXT
415  ENDIF      ! SWEEP
416
417  DO 160 I = 1, 9      ! LOOP THRU ARRAY ELEMENTS
418    IF(IX(I).EQ.IPC(I)LDAT4(I)) - I ! PUT A/D CHANNEL NUMBER IN
419 160  CONTINUE      ! THE ARRAY
420
421    LDAT4(10) = J
422    J = J + 1
423    IF(J.EQ.10)GOTO 200
424    GOTO 60
425  ENDIF
426
427  C -----
428  C THIS PART SORTS EACH ARRAY IN ASCENDING ORDER IF THE LAST
429  C ELEMENT IS NOT A ZERO.

```

## Appendix IV

```
430 C _____
431
432 200 IF(LDAT1(10).GT.0)THEN
433     SIZE - LDAT1(10)
434     CALL BUBBLE_UP(LDAT1,SIZE)
435 ENDIF
436
437 IF(LDAT2(10).GT.0)THEN
438     SIZE - LDAT2(10)
439     CALL BUBBLE_UP(LDAT2,SIZE)
440 ENDIF
441
442 IF(LDAT3(10).GT.0)THEN
443     SIZE - LDAT3(10)
444     CALL BUBBLE_UP(LDAT3,SIZE)
445 ENDIF
446
447 IF(LDAT4(10).GT.0)THEN
448     SIZE - LDAT4(10)
449     CALL BUBBLE_UP(LDAT4,SIZE)
450 ENDIF
451
452 1000 MATRIX - 0
453     RETURN
454     END
```

## Appendix IV

### AIV.22 Analog APSD Software Modules: STAGE\_POSITION Source Code.

```

1  SUBROUTINE STAGE_POSITION(IUT)
2
3  C-----
4  C  THIS ROUTINE ALLOWS THE USER TO SET THE SPEED ,DIRECTION AND
5  C  DISTANCE OF THE STEPPER MOTORS FOR EACH AXIS ON THE CONTROLLER
6  C  FOR EACH LASER IN THE MATRIX EXPERIMENT.
7  C-----
8
9  CHARACTER*10 VEL,ACC,DIST,DUMMY,FILENAME*10
10
11  CHARACTER PORT*10,PORT1*10,LASER*1,YES*1,AXIS*1,MSG*12
12  CHARACTER MSG1*80
13  LOGICAL T
14
15  DIMENSION VEL(4),ACC(4),DIST(4),DUMMY(4)
16
17
18  IUT = 99
19  CALL GETPORT(PORT,I)
20  OPEN(IUT,FILE=PORT,STATUS='NEW')
21
22  C-----
23
24  5  FORMAT(12(A10))
25  10  FORMAT(A80)
26  20  FORMAT(80('-'))
27  30  FORMAT(80(' '))
28  31  FORMAT(' THE FOLLOWING SETTINGS WILL DEFINE STAGE
29  .POSITIONS FOR EACH LASER')
30  32  FORMAT(10X,'1. POSITION STAGES FOR SAMPLE DATA COLLECTION')
31  34  FORMAT(10X,'2. POSITION STAGES FOR THE SPECTRUM ANALYSER')
32  36  FORMAT(10X,'3. EXIT')
33
34
35  40  FORMAT(10X,'ENTER LASER NUMBER.. ( 1 - 4 ): ',5)
36  50  FORMAT(10X,'LASER #',I2,15X,'AXIS #',I2,15X,'CONTROLLER #1')
37  60  FORMAT(10X,'FOR AXIS ',I2,20X,'LASER #',I2)
38  70  FORMAT(10X,'1. ENTER THE VELOCITY: ',5)
39  80  FORMAT(10X,'2. ENTER THE ACCELERATION: ',5)
40  90  FORMAT(10X,'3. ENTER THE DISTANCE ( + OR - ): ',5)
41  100  FORMAT(10X,I2,' VELOCITY: - ',A10)
42  110  FORMAT(10X,I2,' ACCELERATION: - ',A10)
43  120  FORMAT(10X,I2,' DISTANCE IN STEPS: - ',A10)
44  130  FORMAT(24X,'ENTER AXIS NUMBER FOR CORRECTION:')
45  140  FORMAT(24X,' ENTER RETURN TO CONTINUE')
46  150  FORMAT(20X,' < A > TO APPLY OR < E > TO EXIT')
47  160  FORMAT(30X,5)
48  170  FORMAT(24X,' ENTER RETURN FOR NEXT LASER')
49  180  FORMAT(10X,' WHICH SERIAL PORT IS ATTACHED TO THE CONTROLLER: ',
50  .5)
51  190  FORMAT(10X,'ENTER AXIS NUMBER YOU WANT TO TRY: ',5)

```

## Appendix IV

```

52 200 FORMAT(20X,'PRESS ENTER WHEN YOUR READY OR < 99 > TO EXIT')
53 210 FORMAT(10X,'1. THE MOTION IS CORRECT ')
54 220 FORMAT(10X,'2. CHANGE THE DATA FOR THIS AXIS')
55 230 FORMAT(10X,'3. EXIT')
56 240 FORMAT(23X,'DOES THE LASER NEED OPTICS MOVED IN')
57 250 FORMAT(23X,'ORDER TO BE CALIBRATED < Y or N >')
58 260 FORMAT(10X,'THE PRESENT PORT DEFINED FOR OUTPUT IS:',A10)
59 270 FORMAT(29X,'PRESS RETURN IF CORRECT')
60 280 FORMAT(30X,'PRESS < 1 > TO CHANGE')
61
62 C-----
63 C HERE THE USER IS ASKED TO SELECT THE TYPE OF STAGE SETTING FOR
64 C EACH LASER THAT IS TO BE SAVED.
65 C EACH LASER REQUIRES THAT THE OPTICS CHANNEL LIGHT TO THE
66 C MODULATORS FOR CALIBRATION AND TO THE SAMPLE FOR DATA
67 C COLLECTION. THESE SETTINGS WILL OFTEN BE DIFFERENT. TO MAKE
68 C THINGS EASY I AM SAVING THE TWO TYPES OF SETTINGS IN DIFFERENT
69 C FILES. SAMPLE.DAT FOR LIGHT TO THE SAMPLE. MOD.DAT FOR LIGHT
70 C TO THE MODULATORS..
71 C-----
72
73
74 290 WRITE(IUT,30)
75 WRITE(IUT,31)
76 WRITE(IUT,30)
77 WRITE(IUT,32)
78 WRITE(IUT,34)
79 WRITE(IUT,36)
80 WRITE(IUT,30)
81 READ(IUT,'(A)',ERR=300)LASER ! READ TYPE OF STAGE SETTINGS
82
83 IF(LASER.EQ.'1')FILENM = 'SAMP'
84 IF(LASER.EQ.'2')FILENM = 'SPECTRUM'
85 IF(LASER.EQ.'3')GOTO 1000 ! USER WANTS TO EXIT
86
87 300 WRITE(IUT,30)
88 WRITE(IUT,30)
89 WRITE(IUT,140)
90 WRITE(IUT,40) ! ENTER LASER NUMBER
91
92 READ(IUT,'(A)',ERR=300)LASER ! MAXIMUM OF 4 LASERS NOW..
93
94 IF(LASER.EQ.'')GOTO 1000 ! USER WANTS TO EXIT
95
96 IF(LASER.GT.CHAR(48).AND.LASER.LT.CHAR(53))GOTO 310
97 GOTO 300
98
99 310 READ(LASER,'(I4)',ERR=300)IL ! LASER INTEGER VALUE
100
101 IF(IX = 0) ! INITIALIZE DATA TYPE FLAGS
102
103 C-----
104 C THIS CHECKS TO SEE IF THE FILE EXISTS, SETS A FLAG THEN OPENS IT
105 C-----

```

# Appendix IV

```

106
107      IF(IUSED.EQ.0)THEN
108          IUSED - 1          ! SET FLAG NOT TO COME BACK HERE
109          INQUIRE(FILE=FILENM,EXIST = T) ! ASK IF FILE EXISTS
110
111          OPEN(2,FILE = FILENM,ACCESS='DIRECT',FORM='FORMATTED',
112              .RECL=120,STATUS = 'UNKNOWN',SHARED,ERR = 1000)
113
114          READ(2,REC = 9,FMT = '(A10)',ERR = 315)PORT1
115          ENDP
116
117      C-----
118      C   HERE I WRITE BACK TO THE USER THE LAST ENTRY FOR THE AXIS FOR
119      C   THIS LASER. IF THE FILE EXISTS I READ IT FROM THERE.
120      C-----
121
122      315  IF(IJUMP.EQ.1)GOTO 370 ! DONT READ
123
124          IF(.NOT.T)GOTO 370      ! DONT READ IF NO FILE
125
126          READ(2,REC = 11,FMT = 5,ERR = 370)VEL(1),ACC(1),DIST(1),
127              DUMMY(1),VEL(2),ACC(2),DIST(2),DUMMY(2),VEL(3),ACC(3),
128              DIST(3),DUMMY(3)
129
130      C   IF(VEL(1).EQ.' '.OR.ACC(1).EQ.' '.OR.DIST(1).EQ.' '.OR
131      C   .NCHAR(VEL(1)).EQ.0.OR.NCHAR(ACC(1)).EQ.0)THEN
132      C       IJUMP - 1
133      C       GOTO 370
134      C   ENDP
135
136      GOTO 370
137
138      370  K - 0
139          IJUMP - 0          ! RESET THE JUMP FLAG
140
141          DO 390 I = 1,3      ! LOOP THRU THE INPUTS AND DISPLAY
142              ! THE FILE SETTINGS FOR THIS LASER
143              WRITE(IUT,40)I,IL
144              WRITE(IUT,30)
145              K - K + 1
146              WRITE(IUT,100)K,VEL(I)      ! VELOCITY
147              K - K + 1
148              WRITE(IUT,110)K,ACC(I)      ! ACCELERATION
149              K - K + 1
150              WRITE(IUT,120)K,DIST(I)      ! DISTANCE OF TRAVEL
151              WRITE(IUT,20)
152
153      390  CONTINUE
154
155
156          IF (KUM_BACK.EQ.1)GOTO 510
157
158      C-----
159      C   HERE THE USER IS ASKED IF THE DATA IS CORRECT. SHOULD THERE BE

```

# Appendix IV

```

160 C   CHANGES, DOES THE USER WISH TO APPLY THE DATA TO THE STEPPER
161 C   MOTOR CONTROLLERS OR JUST EXIT.
162 C   _____
163
164 400  WRITE(IUT,30)
165      WRITE(IUT,130)
166      WRITE(IUT,140)
167      WRITE(IUT,150)
168      WRITE(IUT,160)
169
170      READ(IUT,'(A1)',ERR - 400)YES
171
172      IF(YES.EQ.' ')THEN
173          ! IF THIS IS CALIBRATION DATA THEN RECORD
174          IREC - IL      ! OR ELSE ITS JUST THE LASER NUMBER
175
176
177      IF(IPIX.EQ.1)THEN      ! SAVE THE NEW DATA IF FLAG IS SET - 1
178
179          WRITE(2,REC - IREC,PMT - 5)VEL(1),ACC(1),DIST(1),
180          DUMMY(1),VEL(2),ACC(2),DIST(2),DUMMY(2),VEL(3),ACC(3),
181          DIST(3),DUMMY(3)
182
183          IPIX - 0          ! RESET THE SAVE DATA FLAG
184          ENDP
185          GOTO 300          ! JUMP TO TOP OF ROUTINE
186
187      ENDP
188
189
190      IF(YES.EQ.'A' OR YES.EQ.'a')GOTO 500
191      IF(YES.EQ.'E' OR YES.EQ.'e')GOTO 1000
192
193 C   _____
194 C   HERE THE USER HAS SELECTED ONE OF THE LASER TO GIVE NEW
195 C   SETTINGS POR. I LOOP THRU THE INPUTS THEN SHOW THEM BACK
196 C   _____
197
198      IF(YES.EQ.'1' OR YES.EQ.'2' OR YES.EQ.'3')THEN
199
200          READ(YES,'(BN,14)',ERR-1000)I
201
202          IPIX - 1          ! FLAG THAT A CHAGE IS TO BE MADE
203
204
205          WRITE(IUT,30)
206          WRITE(IUT,30)
207          WRITE(IUT,30)IL,1
208          WRITE(IUT,30)
209          WRITE(IUT,70)
210          READ(IUT,'(A10)',ERR - 370)VEL(I)
211          WRITE(IUT,30)
212          WRITE(IUT,80)
213          READ(IUT,'(A10)',ERR - 370)ACC(I)

```

# Appendix IV

```

214      WRITE(IUT,30)
215      WRITE(IUT,90)
216      READ(IUT,'(A10)',ERR = 370)DIST(I)
217      WRITE(IUT,30)
218      WRITE(IUT,20)
219
220      GOTO 370      ! DISPLAY INPUTS BACK TO THE USER
221      ENDIF
222
223      GOTO 400      ! INPUT WAS BAD DO AGAIN
224
225      C-----
226      C   THIS PART ALLOWS THE USER TO APPLY THE INPUT BEFORE IT IS SAVED
227      C   TO FILE.  HERE I GET THE SERIAL PORT NUMBER FROM THE USER
228      C   THEN INSTRUCT THE USER TO PRESS ENTER TO HAVE THE DATA SENT
229      C   TO THE SPECIFIC PORT.
230      C-----
231
232      500  WRITE(IUT,30)
233
234      IF(INUSE.EQ.0)THEN
235          INUSE = 1
236          IF(PORT1.EQ.' ' .OR. ICHAR(PORT1(1:1)).EQ.0)THEN
237              505  WRITE(IUT,180)
238              READ(IUT,'(A10)',ERR=500)PORT1
239          ELSE
240              WRITE(IUT,30)
241              WRITE(IUT,260)PORT1
242              WRITE(IUT,270)
243              WRITE(IUT,280)
244              WRITE(IUT,30)
245
246              READ(IUT,'(A1)',ERR = 400)YES
247
248              IF(YES.EQ.'1')GOTO 505
249              IF(YES.NE.' ')GOTO 505
250          ENDIF
251
252          WRITE(*,' PORT1 = ',PORT1)
253
254          OPEN(3,FILE=PORT1,CARRIAGECONTROL = 'NONE',STATUS='NEW')
255          WRITE(3,'(A9)'E MIN STD '      ! WAKE UP CONTROLLER
256
257          WRITE(2,REC = 9,FMT = '(A10)',ERR = 510)PORT1
258
259          ENDIF
260
261
262      C-----
263      C   HERE THE AXIS NUMBER OR STAGE IS REQUESTED FROM THE USER
264      C   AN INPUT OF NOTHING JUST REDISPLAYS THE CURRENT SETTINGS
265      C-----
266
267      510  WRITE(IUT,30)

```

# Appendix IV

```

268      WRITE(TUT,140)
269      WRITE(TUT,190)
270
271      READ(TUT, '(A1)',ERR=510)AXIS
272
273      IF(AXIS.EQ.'1'.OR.AXIS.EQ.'2'.OR.
274      .AXIS.EQ.' '.OR.AXIS.EQ.'3')GOTO 520
275      GOTO 510
276
277 520   KUM_BACK = 0
278
279  C-----
280  C   HERE THE STEPPER MOTORS ARE MOVED FROM THEIR HOME POSITION THE
281  C   VELOCITY, ACCELERATION AND DISTANCE SPECIFIED BY THE ABOVE INPUT.
282  C-----
283
284      READ(AXIS, '(BN,4)')IX      ! CONVERT CHARACTER TO INTEGER
285
286  C-----
287
288      IF(IX.EQ.0)GOTO 370          ! IF THE USER PRESSED RETURN W/OUT
289                                  ! AN INPUT I GO REDISPLAY THE
290                                  ! CURRENT SETTINGS
291  C-----
292  C   HERE I MAKE SURE THAT THERE IS DATA PRIOR TO WRITING TO THE PORT
293  C   IF NOT I NOTIFY THE USER AND TRY AND GO GET SOME...
294  C-----
295
296      IF((VEL(IX).EQ.' '.OR.DIST(IX).EQ.' ')THEN
297          WRITE(TUT,*) 'SORRY THERE IS NO DATA DEFINED FOR THIS AXIS...'
298          GOTO 370
299      ENDIF
300
301
302      WRITE(0, '(A5)AXIS//ST1 '
303      WRITE(*,*)TURN ON THE MOTOR ',AXIS//ST1 '
304
305  C-----
306  C   THIS IS A DUMMY THAT LETS THE USER GET HARDWARE READY PRIOR TO
307  C   MOVING THE ACTUAL STAGES.
308  C-----
309
310 450   FORMAT(A11)
311 451   FORMAT(A12)
312 452   FORMAT(A13)
313 453   FORMAT(A14)
314 454   FORMAT(A15)
315 455   FORMAT(A16)
316 456   FORMAT(A17)
317 457   FORMAT(A18)
318 458   FORMAT(A19)
319 459   FORMAT(A20)
320 460   FORMAT(A21)
321 461   FORMAT(A22)

```

# Appendix IV

```

322 462 FORMAT(A23)
323
324
325
326 WRITE(TUT,30)
327 WRITE(TUT,200)
328
329 READ(TUT,'(A1)')YES
330
331 L = 0
332
333 IF(YES.EQ.' ')THEN
334
335 MSG = AXIS//V//VEL(DX)
336
337 DO 560 I = 1,3
338
339 530 DO 540 J = 1,12
340 IF(MSG(I:J).EQ.' '.OR.MSG(I:J).EQ.'+')GOTO 540
341 L = L + 1
342 MSG1(L:L) = MSG(I:J)
343 540 CONTINUE
344
345 I = I + 1
346 MSG1(I:L) = ' '
347 IF(I.EQ.1)MSG = AXIS//A//ACC(DX)
348 IF(I.EQ.2)MSG = AXIS//D//DIST(DX)
349
350 560 CONTINUE
351
352 L = L + 1
353 MSG1(L:L + 1) = 'G'
354 L = L + 1
355
356 WRITE(*,'Y MSG1 = ',MSG1(1:L), ' I = ',I)
357
358
359 J = L - 10
360 WRITE(*,'Y J = ',J)
361 PAUSE
362 GOTO(570,580,590,600,610,620,630,640,650,660,670,680,690)J
363
364
365 570 WRITE(3,430)MSG1(1:L)
366 GOTO 700
367 580 WRITE(3,431)MSG1(1:L)
368 GOTO 700
369 590 WRITE(3,432)MSG1(1:L)
370 GOTO 700
371 600 WRITE(3,433)MSG1(1:L)
372 GOTO 700
373 610 WRITE(3,434)MSG1(1:L)
374 GOTO 700
375 620 WRITE(3,435)MSG1(1:L)

```

# Appendix IV

```

376      GOTO 700
377 630  WRITE(3,456)MSG1(1:L)
378      GOTO 700
379 640  WRITE(3,457)MSG1(1:L)
380      GOTO 700
381 650  WRITE(3,458)MSG1(1:L)
382      GOTO 700
383 660  WRITE(3,459)MSG1(1:L)
384      GOTO 700
385 670  WRITE(3,460)MSG1(1:L)
386      GOTO 700
387 680  WRITE(3,461)MSG1(1:L)
388      GOTO 700
389 690  WRITE(3,462)MSG1(1:L)
390
391      ENDIF
392
393  C-----
394  C   HERE THE USER CAN CHECK TO SEE IF THE MOVEMENT WAS CORRECT.
395  C   IF NOT THE DATA CAN BE CHANGED.
396  C-----
397
398 700  WRITE(TUT,30)
399      WRITE(TUT,210)      ! STAGE MOVEMENT CORRECT
400      WRITE(TUT,220)      ! CHANGE THE MOTION
401      WRITE(TUT,230)      ! EXIT
402      WRITE(TUT,30)
403
404      READ(TUT,'(A1)',ERR=700)YES
405
406  C-----
407  C   HERE I SEND THE STAGE BACK TO ITS HOME POSITION < H > MEANS
408  C   REVERSE DIRECTION AND < G > TELLS IT TO GO THE SAME DISTANCE
409  C   AND VELOCITY.
410  C-----
411
412      WRITE(3,'(A4)'H G '
413      WRITE(3,'(A5)'AXIS/STO '
414
415  C-----
416
417      IF(YES.EQ.'1')THEN    ! THE DATA IS CORRECT WRITE IT TO FILE
418
419          IREC = IL        ! OR ELSE ITS JUST THE LASER NUMBER
420
421          WRITE(2,REC - IREC,PMT = 5)VEL(1),ACC(1),DIST(1),
422          .DUMMY(1),VEL(2),ACC(2),DIST(2),DUMMY(2),VEL(3),ACC(3),
423          .DIST(3),DUMMY(3)
424
425          KUM_BACK = 1
426          GOTO 370        ! GOTO GET MORE DATA FOR THESE AXIS
427
428  C-----
429  C   HERE THE USER DOES NOT LIKE THE STAGE MOVEMENT AND WOULD LIKE TO

```

# Appendix IV

```

430 C   MAKE AN ADJUSTMENT. I SEND THE USER BACK UP IN THE ROUTINE WHERE
431 C   THE CORRECTIONS ARE MADE.
432 C _____
433
434     ELSEIF(YES.EQ.'2')THEN
435
436         I - IX           ! PLACE THE AXIS NUMBER TO CORRECT IN I
437
438         GOTO 370         ! GOTO BACK THRU THE ROUTINE.
439
440 C _____
441 C   HERE THE USER WANTS TO EXIT.
442 C _____
443
444     ELSEIF(YES.EQ.'3')THEN
445         GOTO 300         ! GO TO THE EXIT CLOSE THE FILES...
446
447     ELSE
448
449         GOTO 700         ! BAD INPUT GO BACK TO THE READ...
450
451     ENDIF
452
453 1000 WRITE(3, '(A2)')P '      ! TAKE CONTROLLER OFF LINE.
454      CLOSE(3)
455      CLOSE(2)
456
457      RETURN
458      END

```

# Appendix IV

## AIV.23 Analog APSD Software Modules: TEK3 Source Code.

```

1      SUBROUTINE TEK3(TYPE,SAMP,AGENT,CONC,LAMDA)
2
3      C-----
4      C   THIS ROUTINE WILL TAKE REAL TIME MULLER MATRIX DATA AND PRESENT IT
5      C   IN GRAPHIC FORM ON A TEKTRONIX COMPUTER.
6      C-----
7
8      CHARACTER E,PORT*5,A*5,A1*5,SEGCNT*3,TEXT*30,TIC1*4,COL*2,LET*1
9      CHARACTER SAMP*20,AGENT*20,CONC*15,LAMDA*15,TIC*3,NUM*2,CP*2
10     CHARACTER TEXT1*6,SCNT*1,ANS*12,SEG*3,POINT*6,KEY,FILENM*20
11
12     REAL RARM,REND,RINCR,RDAT
13
14     INTEGER X,Y,MATRIX,TYPE
15
16     DIMENSION TIC(18),TIC1(21),NUM(16),COL(16),CP(16),TEXT1(2)
17     DIMENSION LDAT1(18),LDAT2(18),LDAT3(18),LDAT4(1)
18
19
20
21     DATA TIC /'10','20','30','40','50','60','70','80','90','100','110',
22     '120','130','140','150','160','170','180'/
23
24     DATA TIC1 /+1.0,+ .9,+ .8,+ .7,+ .6,+ .5,+ .4,
25     '+ .3,+ .2,+ .1, 0.0,- .1,- .2,- .3,- .4,- .5,
26     '- .6,- .7,- .8,- .9,-1.0/
27
28     DATA NUM/'11','12','13','14','21','22','23','24',
29     '31','32','33','34','41','42','43','44'/
30     DATA COL /'2','1','4','1','2','1','2','1','4','1','1','1',
31     '1','1','2'/
32
33     DATA CP /'T','O','S','S','S','Q','C','Y','T','I',
34     'I','I','I'/
35
36     DATA TEXT1 / 'EXIT',' ALL'/
37
38
39     E = CHAR(27)
40     IFILECNT = 0
41     IFLG = 0
42     ITOGH = 0
43     ITOCV = 0
44
45     C-----
46     C           ***** TEST DATA *****
47     C   SAMP = 'GREEN PAINT'
48     C   AGENT = 'GB'
49     C   CONC = '.003 Mg/m3'
50     C   LAMDA = '.094 um'
51     C   WRITE(*,E//'.10'

```

# Appendix IV

```

32  C_____
33
34  WRITE(*,"E/LV0"
35  WRITE(*,"E/LZ"
36
37  1  FORMAT (A12)
38
39
40
41
42
43  C_____
44  C  THIS PART DRAWS THE GRAPH BOX. COLOR RED.
45  C_____
46
47  ISEC - 1
48
49  IF(TYPE.EQ.1)GOTO 5      ! GRAPH ALREADY EXISTS
50
51  CALL INTRPT(ISEC,SECCNT)
52
53  WRITE(*,"E/SE//SECCNT
54  WRITE(*,"E/MLZ"
55  X - 145
56  Y - 3100
57  CALL HTY(X,Y,A)
58  WRITE(*,"E/LP//A
59  X - X + 3800
60  CALL HTY(X,Y,A)
61  WRITE(*,"E/LG//A
62  Y - Y - 2400
63  CALL HTY(X,Y,A)
64  WRITE(*,"E/LG//A
65  X - X - 3800
66  CALL HTY(X,Y,A)
67  WRITE(*,"E/LG//A
68  Y - Y + 2400
69  CALL HTY(X,Y,A)
70  WRITE(*,"E/LG//A
71  WRITE(*,"E/FC"
72
73  C_____
74  C  THIS DRAWS THE TEXT IN THE BOX. SAMPLE,AGENT, CONC, WAVELENGTH
75  C  START,END AND INCREMENT POINTS
76  C  THIS FIRST ONE WRITES THE SAMPLE
77  C_____
78
79  5  ISEC - ISEC + 1
80
81
82  CALL INTRPT(ISEC,SECCNT)
83  WRITE(*,"E/SC//SECCNT
84  WRITE(*,"E/SE//SECCNT
85
86  WRITE(*,"E/MTT"

```

# Appendix IV

```

106      X - 600
107      Y - 2950
108      CALL HTY(X,Y,A)
109      WRITE(*,*)E//LP//A
110      TEXT - 'SAMPLE: '//SAMP
111      WRITE(*,*)E//LTA;//TEXT
112
113      C-----
114      C   THIS WRITES THE AGENT TYPE
115      C-----
116
117      X - 2300
118      CALL HTY(X,Y,A)
119      WRITE(*,*)E//LP//A
120      TEXT - 'AGENT: '//AGENT
121      WRITE(*,*)E//LTA;//TEXT
122
123      C-----
124      C   THIS WRITE THE AGENT CONCENTRATION
125      C-----
126
127      X - 600
128      Y - 750
129      CALL HTY(X,Y,A)
130      WRITE(*,*)E//LP//A
131      TEXT - 'CONC: '//CONC
132      WRITE(*,*)E//LTA;//TEXT
133
134      C-----
135      C   THIS WRITES THE LASER WAVELENGTH
136      C-----
137
138      X - 2300
139      Y - 750
140      CALL HTY(X,Y,A)
141      WRITE(*,*)E//LP//A
142      TEXT - 'WAVELENGTH: '//LAMDA
143      WRITE(*,*)E//LTA;//TEXT
144      WRITE(*,*)E//SC
145
146
147      IF(TYPE.EQ.1)GOTO 1010
148
149      C-----
150      C   THIS WRITES THE BAR GRAPH IN THE CENTER .
151      C-----
152
153
154      ISEC - 1000
155
156      CALL INTRPT(ISEC,SECONT)
157      WRITE(*,*)E//SC//SECONT
158      WRITE(*,*)E//SE//SECONT
159

```

# Appendix IV

```

160      WRITE(*,"E/ML15'
161      X - 245
162      Y - 1900
163
164      CALL HTY(X,Y,A)
165      A1 - A
166      WRITE(*,"E/LP//A
167      X - 3045
168      CALL HTY(X,Y,A)
169      WRITE(*,"E/LC//A
170
171      C-----
172
173      C   NOW THE TICs ARE ADDED TO THE LINE
174      C-----
175
176      X - 245
177      Y - Y - 25
178      CALL HTY(X,Y,A)
179      WRITE(*,"E/LP//A
180      Y - Y + 50
181      CALL HTY(X,Y,A)
182      WRITE(*,"E/LC//A      ! THE FIRST TIC 50 PIXELS
183
184      X - X - 5
185      Y - Y - 90
186      CALL HTY(X,Y,A)
187      WRITE(*,"E/LP//A
188      WRITE(*,"E/MC:A45'      ! CHANGE SIZE OF TEXT
189      WRITE(*,"E/LT:0'      ! PLACE 0 UNDER FIRST TIC
190
191      X - 245
192      Y - Y + 77
193
194      C-----
195
196      DO 40 I=1,18
197
198      DO 20 II=1,9
199
200      X - X + 20
201      Y - Y - 25
202      CALL HTY(X,Y,A)
203      WRITE(*,"E/LP//A
204      Y - Y + 25
205      CALL HTY(X,Y,A)
206      WRITE(*,"E/LC//A
207
208      20  CONTINUE
209
210      C-----
211      C   THIS PART PLACE THE LARGER TIC MARKS AND LABELS THEM
212      C-----
213

```

# Appendix IV

```

214      X - X + 20
215      Y - Y - 37
216      CALL HTY(X,Y,A)
217      WRITE(*,"E//LP//A
218      Y - Y + 50
219      CALL HTY(X,Y,A)
220      WRITE(*,"E//LG//A      ! THE TIC MARKS AT 10 DEG INTERVALS
221
222      X - X - 5
223      Y - Y - 90
224      CALL HTY(X,Y,A)
225      WRITE(*,"E//LP//A
226      WRITE(*,"E//LT3//TIC()
227      Y - Y + 77
228      X - X + 5
229
230 40    CONTINUE
231      WRITE(*,"E//SC
232
233 C-----
234 C      THIS PLACES THE VERTICAL TICS AND TEXT OF THE GRAPH
235 C-----
236
237      ISEG - ISEG + 1
238      X - 150
239      Y - 2890
240
241      CALL INTRPT(ISEG,SEGCNT)
242
243      WRITE(*,"E//SE//SEGCNT
244
245      DO 60 I = 1, 21
246
247      WRITE(*,"E//ML7"
248      CALL HTY(X,Y,A)
249      WRITE(*,"E//LP//A
250      WRITE(*,"E//LT4//TIC1()
251
252      WRITE(*,"E//ML15"
253      X - X + 65
254      Y - Y + 10
255      CALL HTY(X,Y,A)
256      WRITE(*,"E//LP//A
257
258      X - X + 30
259      CALL HTY(X,Y,A)
260      WRITE(*,"E//LG//A
261
262      X - X - 30
263      Y - Y - 10
264
265 C-----
266 C      YOU DONT NEED TIC MARKS AFTER THE LAST NUMBER
267 C-----

```

# Appendix IV

```

268
269      IF(I.EQ.21)GOTO 60
270
271 C-----
272 C   THIS DRAWS 9 TIC MARKS IN BETWEEN THE LARG ONES
273 C-----
274
275      DO 50 J - 1,9
276
277      CALL HTY(X,Y,A)
278      WRITE(*,"E/1P"//A
279
280      X - X + 25
281      CALL HTY(X,Y,A)
282      WRITE(*,"E/1P"//A
283      Y - Y - 10
284      X - X - 25
285 50  CONTINUE
286      Y - Y - 10
287      X - X - 65
288
289 60  CONTINUE
290
291      WRITE(*,"E/1SC"
292
293 C-----
294 C   AT THIS POINT WE MAKE 16 BOXES OF DIFFERENT COLORS THAT REPRESENT
295 C   THE MULLER MATRIX. TWO OF THE BOXES WILL BE THE SAME COLOR WHITE
296 C   BECAUSE BLACK IS COLOR 16 AND THAT IS THE LIMIT OF A 4111. BUT
297 C   SINCE THESE ARE LINES I WILL JUST CHANGE THE STYLE OF LINE 16.
298 C
299 C   THE MATRIX ELEMENTS ARE DEFINED AS SEGMENT NUMBERS 10 - 25
300 C-----
301
302 65  L - 0
303      K - 0
304      X - 3000
305      Y - 600
306
307      ISEG - 9
308
309      DO 80 I - 1,4          ! ROW LOOP
310      DO 70 J - 1,4          ! COLUMN LOOP
311          L - L + 1          ! COLOR INDEX COUNTER
312          ISEG - ISEG + 1    ! SEGMENT COUNTER
313          CALL INTRPT(ISEG,SEGCNT) ! CONVERT INTEGER TO TEK
314
315          WRITE(*,"E/1SE"//SEGCNT ! BEGIN THE SEGMENT
316
317 C-----
318 C   THIS SETS THE BLOCK COLOR
319 C-----
320
321      IF(I.EQ.4.AND.J.EQ.4)WRITE(*,"E/1MV1" ! DASHED LINE ON LAST BOX

```

# Appendix IV

```

322
323      IF(CP(L).EQ.'Q')CP(L) = CHAR(39) ! DEFINE THE COLOR
324
325      WRITE(*,"E//MP//CP(L)          ! WRITE THE COLOR TO TERM
326
327  C-----
328
329      CALL HTY(X,Y,A)                ! CONVERT ORIGIN VECTOR
330
331      WRITE(*,"E//LP//A//1'          ! AND WRITE IT TO TERMINAL
332
333
334      X = X + 150                    ! INCREMENT THE X BY 150
335      CALL HTY(X,Y,A)                ! CONVERT IT TO TEK CHARA
336      WRITE(*,"E//LG//A              ! DRAW FROM LAST VECTOR
337
338      Y = Y - 100                    ! DECREMENT THE Y BY 100
339      CALL HTY(X,Y,A)                ! CONVERT TO TEK CHARACTER
340      WRITE(*,"E//LG//A              ! DRAW FROM LAST VECTOR
341
342      X = X - 150                    ! DECREMENT X BY 150
343      CALL HTY(X,Y,A)                ! CONVERT TO TEK CHARACTER
344      WRITE(*,"E//LG//A              ! DRAW FROM LAST VECTOR
345      WRITE(*,"E//LE'                ! FILL THE PANEL
346
347  C-----
348  C   THIS PUTS THE TEXT NUMBER IN THE BOX
349  C-----
350
351      K = K + 1                      ! ELEMENT COUNTER USED FOR
352                                     ! PLACING THE TEXT. IF
353  C   IF(K.LT.10)THEN                ! THE TEXT IS 1 CHARACTER
354  C     X = X + 60                    ! THEN X IS INCREMENTED BY
355  C     Y = Y + 25                    ! 60 AND Y BY 25. IF THE
356  C   ELSE                            ! TEXT IS 2 CHARACTERS
357  C     X = X + 30                    ! X IS INCREMENTED BY 30
358  C     Y = Y + 25                    ! SO THAT IT IS CENTERED.
359  C   ENDIF
360
361      WRITE(*,"E//MCB4C4'            ! THIS IS THE TEXT SIZE
362
363      CALL HTY(X,Y,A)                ! CONVERT X,Y TO TEK CHARA
364      WRITE(*,"E//LP//A              ! SET THE ORIGIN
365      WRITE(*,"E//MT//COL(K)         ! SET THE BOX COLOR
366      WRITE(*,"E//LT2//NUM(K)        ! WRITE THE TEXT
367
368      WRITE(*,"E//SC'                ! CLOSE THE SEGMENT
369
370  C   IF(K.LT.10)THEN                ! INCREMENT X AND Y FOR
371  C     X = X + 140                    ! THE NEXT BOX IN THE
372  C     Y = Y + 75                    ! ROW.
373  C   ELSE
374  C     Y = Y + 75
375  C     X = X + 155

```

# Appendix IV

```

376 C   ENDIF
377
378
379 70  CONTINUE          ! END COLUMN LOOP
380
381     X - 3000          ! GO TO START POSITION
382     Y - Y - 150      ! FOR THE NEXT ROW
383
384 80  CONTINUE          ! END ROW LOOP
385
386 C-----
387 C   HERE I AM PROVIDING TWO BUTTONS FOR THE USER. THE FIRST IS "EXIT"
388 C   AND THE OTHER IS "ALL ". WHICH IF SELECTED WILL INDICATE TO THE
389 C   USER THAT ALL INCOMING VECTOR INFORMATION IS TO BE DISPLAYED.
390 C   THIS MEANS THAT IN EACH SWEEP OF THE SAMPLE STAGE 9 ELEMENTS WILL
391 C   BE SIMULTANEOUSLY DRAWN TO THE BAR GRAPH WHICH WILL DIRECTLY
392 C   REPRESENT 9 MATRIX ELEMENTS PROVIDING DATA FOR THAT PARTICULAR
393 C   POLARIZING POSITION.
394 C
395 C   THESE SEGMENTS WILL BE 40 AND 41 RESPECTIVELY.
396 C-----
397
398     ISEG - 39
399     X - 2000          ! X ORIGIN OF EXIT BOX
400     Y - 650          ! Y ORIGIN OF EXIT BOX
401     K - 0
402     WRITEC,"E//MVO"    ! SOLID LINES
403
404     DO 100 J - 1,2
405
406     ISEG - ISEG + 1    ! INCREMENT SEGMENT CNTR
407     CALL INTRPT(ISEG,SEGCNT) ! CONVERT INTEGER TO TEK
408
409     WRITEC,"E//50//SEGCNT" ! BEGIN THE SEGMENT
410
411 C-----
412 C   THIS SETS THE BLOCK COLOR
413 C-----
414
415     IF(J.EQ.2)THEN      ! IF "ALL" BOX THEN
416         WRITEC,"E//MPS" ! COLOR IT BLUE
417     ELSE                ! OR ELSE IF ITS THE
418         WRITEC,"E//MP"  ! "EXIT" BOX THEN COLOR
419     ENDIF               ! IT RED.
420
421
422     CALL HTY(X,Y,A)      ! CONVERT VECTOR TO TEK
423     WRITEC,"E//LP//A//1" ! SET THE ORIGIN
424
425     X - X + 400          ! INCREMENT THE X BY 400
426     CALL HTY(X,Y,A)      ! CONVERT TO TEK CHARACTER
427     WRITEC,"E//LG//A"    ! DRAW FROM LAST VECTOR
428
429     Y - Y - 100          ! DECREMENT THE Y BY 100

```

# Appendix IV

```

430      CALL HTY(X,Y,A)          ! CONVERT TO TEK CHARACTER
431      WRITE(*,"E//LG'//A      ! DRAW FROM LAST VECTOR
432
433      X - X - 400              ! DECREMENT X BY 400
434      CALL HTY(X,Y,A)          ! CONVERT TO TEK CHARACTER
435      WRITE(*,"E//LG'//A      ! DRAW FROM LAST VECTOR
436      WRITE(*,"E//LE'         ! FILL THE BOX WITH COLOR
437
438      C _____
439      C  THIS PUTS THE TEXT NUMBER IN THE BOX
440      C _____
441
442      X - X + 50               ! X,Y ARE SET TO PLACE
443      Y - Y + 25               ! THE TEXT ORIGIN
444      K - K + 1               ! ARRAY COUNTER
445
446      WRITE(*,"E//MCB4C4:'     ! THIS IS TEXT SIZE
447
448      CALL HTY(X,Y,A)          ! ORIGIN INTEGER TO TEK
449      WRITE(*,"E//LP'//A       ! SET THE ORIGIN
450      WRITE(*,"E//MT1'         ! TEXT COLOR WHITE
451      WRITE(*,"E//LT6'//TEXT1(K) ! WRITE THE TEXT
452
453      WRITE(*,"E//SC'          ! CLOSE THE SEGMENT
454
455      X - X + 400              ! SET UP FOR NEXT BOX
456      Y - Y + 75
457      100 CONTINUE            ! LOOP FOR NEXT BOX
458
459      C _____
460      C  THIS PART ACTIVATES A SUBROUTINE TO PLACE A GIN DEVICE FOR THE
461      C  MOUSE.
462      C _____
463
464      X - 2350
465      Y - 600
466      IFLAG - 1
467
468      CALL GIN(X,Y,IFLAG,IMODE,ITYPE IGIN,IPORT)
469
470      C _____
471      C  THIS DRAWS A BLUE PANEL IN THE LOWER LEFT PART OF THE SCREEN
472      C  WITH A RED BORDER. THIS WILL BE USED AS THE BACKGROUND FOR THE
473      C  DIALOG AREA WHICH WILL SIT ON TOP OF THIS PANEL.
474      C _____
475
476
477      ISEG - 1010
478      CALL INTRPT(ISEG,SEGCNT)
479      WRITE(*,"E//SE'//SEGCNT
480      WRITE(*,"E//MPS'
481      WRITE(*,"E//MLX'
482      X - 0
483      Y - 0

```

# Appendix IV

```

484      CALL HTY(X,Y,A)
485      WRITE(*,*)E//LP//A//1'
486      X = 0
487      Y = 500
488      CALL HTY(X,Y,A)
489      WRITE(*,*)E//LG//A
490      X = 1950
491      Y = 500
492      CALL HTY(X,Y,A)
493      WRITE(*,*)E//LG//A
494      X = 1950
495      Y = 0
496      CALL HTY(X,Y,A)
497      WRITE(*,*)E//LG//A
498      WRITE(*,*)E//SC'
499
500  C -----
501  C   THIS PART ESTABLISHES THE DIALOG AREA TO BE 40 CHARACTERS IN
502  C   LENGTH, 5 LINES WIDE, WITH ORIGIN IN THE LOWER LEFT.
503  C -----
504
505  112  WRITE(*,*)E//LV0'
506      WRITE(*,*)E//LX00'
507      WRITE(*,*)E//LLS'
508
509  115  WRITE(*,*)E//LZ'
510      WRITE(*,*)E//LCB'
511      WRITE(*,*)E//LN44'
512      WRITE(*,*)E//LV1'
513
514      INU = 1
515      CALL SEE_ELE(SWEEP,MATRIX,INU)
516
517  1000  WRITE(*,*)E//LV0'
518      WRITE(*,*)E//TD'
519      WRITE(*,*)E//LZ'
520      WRITE(*,*)E//LV1'
521
522  1010  RETURN
523
524      END

```

## Appendix IV

### AIV.24 Analog APSD Software Modules: TEK\_INPUTS Source Code.

```

1      SUBROUTINE TEK_INPUTS(NOSAMP,SAMP,AGENT,CONC,START,STOP,
2      .      INC,NAME,DATE,TIME,TEXT)
3
4      C-----
5      C   THIS MOD IS USED FOR ENTERING DATA IF THE USER WANTS TEKTRONIX
6      C   GRAPHICS. IT ALLOWS THE USER TO ENTER ALL SAMPLE AND LASER DATA
7      C   FOR THE MULLER MATRX EXPERIMENT.
8      C-----
9
10
11      CHARACTER TITLE*48,NAME*20,SAMP*20,AGENT*20,CONC*15
12      CHARACTER SAMPLE,CORRECT*2
13      CHARACTER DATE*9,TIME*8,TEXT*80,E*1,A*5,SEG*3
14      CHARACTER SAME*1,START*6,STOP*6,INC*6,POS*6
15
16      INTEGER CHANGE,X,Y,X1,Y1,Y2,Y3
17
18      C-----
19
20      DIMENSION SAMP(10),AGENT(10),CONC(10)
21      DIMENSION START(10),STOP(10),INC(10),POS(10)
22
23      1  FORMAT('1',/111111116,A,/)
24      2  FORMAT(A20)
25      3  FORMAT(A4)
26      4  FORMAT('O')
27      5  FORMAT(A2)
28      6  FORMAT (A)
29      7  FORMAT(I5)
30      8  FORMAT (A10)
31
32
33      E  - CHAR(27)
34      CHANGE - 0
35      ISTART - 1
36      IEXT - 0      ! USER WANTS TO QUIT ...RETURN
37
38      C-----
39      C   THIS PART PLACES A GREEN PANEL SEGMENT 8000 ON THE ENTIRE SCREEN
40      C   FOR A BACKGROUND.
41      C-----
42
43      WRITE(*,*)E//%10'      ! PLACE IN TEK MODE
44
45      ISEG - 8000
46      CALL INTRPT (ISEG,SEG)
47      WRITE(*,*)E//SE//SEG      ! BEGIN THE PANEL 8000
48      WRITE(*,*)E//MP%      ! PANEL COLOR GREEN
49      X - 1
50      Y - 1
51      CALL HTY(X,Y,A)

```

# Appendix IV

```

52      WRITE(*,"E/LP//A      ! SET PANEL ORIGIN
53
54      X - 4095
55      CALL HTY(X,Y,A)
56      WRITE(*,"E/LG//A      ! DRAW BOTTOM OF PANEL
57  c    Y - 3150
58      Y - 3276
59      CALL HTY(X,Y,A)
60      WRITE(*,"E/LG//A      ! DRAW LEFT SIDE OF PANEL
61      X - X - 4094
62      CALL HTY(X,Y,A)
63      WRITE(*,"E/LG//A      ! DRAW TOP OF PANEL
64      WRITE(*,"E/SC        ! CLOSE AND FILL PANEL
65
66  C-----
67  C  THIS IS THE FIRST LINE THAT IS REQUIRED BY THE USER.
68  C  NAME, DATE, TIME AND NUMBER OF SAMPLES.
69  C-----
70
71
72      CALL TIME4(DATE,TIME)      ! SYSTEM TIME AND DATE
73      TEXT - '1. NAME'
74      ISEC - 1
75      CALL INTRPT(ISEC,SEC)
76
77      WRITE(*,"E/SE//SEG      ! BEGIN THE SEGMENT
78      WRITE(*,"E/MT4'        ! LINE COLOR WHITE
79      X - 100
80      Y - 3100
81      CALL HTY (X,Y,A)
82      WRITE(*,"E/LP//A      ! SET TEXT ORIGIN
83      WRITE(*,"E/LTY//DATE    ! WRITE THE DATE
84      X - 1000
85      CALL HTY (X,Y,A)
86      WRITE(*,"E/LP//A      ! SET TEXT ORIGIN
87      WRITE(*,"E/LTY//TIME    ! WRITE THE TIME
88
89      WRITE(*,"E/MT2'        ! LINE COLOR RED
90      X - 2000
91      CALL HTY (X,Y,A)
92      WRITE(*,"E/LP//A      ! SET TEXT ORIGIN
93      WRITE(*,"E/LTY//TEXT    ! WRITE THE DATE
94      WRITE(*,"E/SC
95
96  C-----
97  C  THIS PART DRAWS A BOX AT THE BOTTOM OF THE SCREEN FOR THE DIALOG
98  C  AREA INPUTS.
99  C-----
100
101      ISEC - ISEC + 1
102      CALL INTRPT (ISEC,SEC)
103      WRITE(*,"E/SE//SEG      ! BEGIN THE PANEL 8000
104      WRITE(*,"E/ML1'        ! LINE COLOR WHITE
105      WRITE(*,"E/MP8'        ! PANEL COLOR BLUE

```

# Appendix IV

```

106      X - 1
107      Y - 1
108      CALL HTY(X,Y,A)
109      WRITE(*,"E//LP//A//1"      ! SET PANEL ORIGIN
110      X - 4095
111      CALL HTY(X,Y,A)
112      WRITE(*,"E//LG//A          ! DRAW BOTTOM OF PANEL
113      Y - 450
114      CALL HTY(X,Y,A)
115      WRITE(*,"E//LG//A          ! DRAW LEFT SIDE OF PANEL
116      X - X - 4094
117      CALL HTY(X,Y,A)
118      WRITE(*,"E//LG//A          ! DRAW TOP OF PANEL
119      WRITE(*,"E//LE"            ! FILL THE PANEL
120
121      C-----
122      C   THIS DRAWS A LINE AROUND THE TWO DIALOG AREA LINES: COLOR RED
123      C-----
124
125      WRITE(*,"E//ML2"            ! PANEL COLOR RED
126      X - 100
127      Y - 150
128      CALL HTY(X,Y,A)
129      WRITE(*,"E//LP//A          ! SET PANEL ORIGIN
130      X - 3995
131      CALL HTY(X,Y,A)
132      WRITE(*,"E//LG//A          ! DRAW BOTTOM OF PANEL
133      Y - Y + 250
134      CALL HTY(X,Y,A)
135      WRITE(*,"E//LG//A          ! DRAW LEFT SIDE OF PANEL
136      X - 100
137      CALL HTY(X,Y,A)
138      WRITE(*,"E//LG//A          ! DRAW TOP OF PANEL
139      Y - Y - 250
140      CALL HTY(X,Y,A)
141      WRITE(*,"E//LG//A          ! DRAW LEFT SIDE OF PANEL
142
143      C-----
144
145      WRITE(*,"E//MT1"            ! TEXT COLOR WHITE
146
147      TEXT - 'ENTER < E > TO EXIT'
148
149      X - 1600
150      Y - 45
151      CALL HTY(X,Y,A)
152      WRITE(*,"E//LP//A          ! SET TEXT ORIGIN
153      WRITE(*,"E//LTA4//TEXT      ! WRITE THE TEXT
154
155      WRITE(*,"E//FSC"            ! CLOSE AND FILL PANEL
156
157      C-----
158      C   THIS SETS UP THE DIALOG AREA SO THAT THE TEXT AND DATA ENTRY
159      C   ARE ALL WITHIN THE DATA INPUT WINDOW.

```

# Appendix IV

```

160 C _____
161
162 WRITE(*,"E/LV0"      ! DISABLE DIALOG AREA
163 WRITE(*,"E/LZ"       ! CLEAR DIALOG AREA
164 WRITE(*,"E/LL2"      ! DIALOG AREA 2 LINES
165 WRITE(*,"E/LCD1"     ! 65 CHARACTER ALLOWED
166 WRITE(*,"E/ML1"     ! DIALOG TEXT WHITE
167 X - 200
168 Y - 160
169 CALL HTY(X,Y,A)
170 WRITE(*,"E/LX"//A    ! SET TEXT DIALOG ORIGIN
171
172 WRITE(*,"E/LV1"      ! ENABLE DIALOG AREA
173
174
175 C _____
176 C   THIS GETS THE NAME OF THE PERSON RUNNING THE EXPERIMENT
177 C _____
178 C*****
179
180 30  FORMAT(5X,"ENTER YOUR NAME: ",5)  ! GET USERS NAME
181
182 40  WRITE(*,30,ERR=40)
183 READ(*,2,ERR=40) NAME      ! READ EXPERIMENTER NAME
184
185 IF(NAME.EQ.'E'.OR.NAME.EQ.'e')THEN  ! USER WANTS TO EXIT
186   EXIT - 1      ! SET EXIT FLAG
187   GOTO 1000     ! RETURN TO CALLER
188 ENDIF
189
190 C _____
191 C   THIS WRITES THE USER INPUTS TO THE SCREEN PANEL AS A GRAPHIC
192 C _____
193
194 ISEG - 3
195 CALL INTRPT(ISEG,SEG)
196
197 WRITE(*,"E/SK"//SEG      ! DELETE THE SEGMENT
198
199 WRITE(*,"E/SE"//SEG      ! BEGIN THE SEGMENT
200 WRITE(*,"E/MP"          ! PANEL COLOR GRAY
201 WRITE(*,"E/MT1"         ! TEXT COLOR WHITE
202
203 X - 2500
204 Y - 3075
205 CALL HTY(X,Y,A)
206 WRITE(*,"E/LP"//A      ! SET PANEL ORIGIN
207 X - X + 1100
208 CALL HTY(X,Y,A)
209 WRITE(*,"E/LG"//A      ! DRAW BOTTOM OF PANEL
210 Y - Y + 100
211 CALL HTY(X,Y,A)
212 WRITE(*,"E/LG"//A      ! DRAW LEFT SIDE OF PANEL
213 X - X - 1100

```

# Appendix IV

```

214      CALL HTY(X,Y,A)
215      WRITE(*,*)E//LC//A      ! DRAW TOP OF PANEL
216      WRITE(*,*)E//LE      ! FILL THE PANEL
217
218      C-----
219      C   THIS PLACES THE TEXT IN THE PANEL
220      C-----
221
222      X - 2550
223      Y - 3100
224      CALL HTY(X,Y,A)
225      WRITE(*,*)E//LP//A      ! SET THE ORIGIN
226      WRITE(*,*)E//LTA4//NAME      ! WRITE THE NAME
227      WRITE(*,*)E//PSC      ! CLOSE THE SEGMENT
228
229      C-----
230
231      IF (CHANGE.EQ.1) GOTO 500
232
233      C-----
234      C-----
235      C   THIS GETS THE NUMBER OF SAMPLES AND THE NAME OF EACH SAMPLE
236      C-----
237      C-----
238
239      50  FORMAT(SX,' ENTER NUMBER OF SAMPLES < 8 MAX > : ',S)
240      60  WRITE(*,30)
241      READ (*, '(A1)',ERR=60)SAMPLE      ! INPUT NUMBER OF SAMPLES
242
243      IF(SAMPLE.EQ.'E'.OR.SAMPLE.EQ.'e')THEN ! USER WANTS TO EXIT
244          TEXTT - 1      ! SET EXIT FLAG
245          GOTO 1000      ! RETURN TO CALLER
246      ENDIF
247
248      IF(ICHAR(SAMPLE).GT.48.AND.ICHAR(SAMPLE).LT.57)GOTO 65
249      GOTO 60      ! INPUT BAD DO IT AGAIN
250
251      65  READ(SAMPLE,'(I4)',ERR = 60)NOSAMP      ! CONVERT TO INTEGER
252
253      C-----
254      C   THIS DRAWS THE SAMPLE DATA HEADER AND NUMBER OF SAMPLES
255      C-----
256
257      ISEG - 7
258      CALL INTRPT(ISEG,SEG)
259      WRITE(*,*)E//SK//SEG      ! DELETE THE SEGMENT
260      WRITE(*,*)E//SE//SEG      ! BEGIN THE SEGMENT
261
262      X - 1
263      Y - 3025
264      CALL HTY(X,Y,A)
265      WRITE(*,*)E//MLA'      ! LINE COLOR BLUE
266      WRITE(*,*)E//LP//A      ! SET LINE ORIGIN
267

```

# Appendix IV

```

268      X - 4095
269      CALL HTY(X,Y,A)
270      WRITE(*,"E//LG"//A          ! SET LINE END
271
272      WRITE(*,"E//MTZ"            ! TEXT COLOR RED
273
274      X - 2000
275      Y - 2095
276      CALL HTY(X,Y,A)
277      WRITE(*,"E//LP"//A          ! SET TEXT ORIGIN
278      TEXT - "2. "
279      WRITE(*,"E//LT4"//TEXT      ! WRITE THE TEXT
280
281      WRITE(*,"E//MPY"            ! PANEL COLOR GRAY
282      WRITE(*,"E//MT1"            ! TEXT COLOR WHITE
283
284      X - 2500
285      Y - 2070
286      CALL HTY(X,Y,A)
287      WRITE(*,"E//LP"//A          ! SET PANEL ORIGIN
288
289      X - X + 1100
290      CALL HTY(X,Y,A)
291      WRITE(*,"E//LG"//A          ! DRAW BOTTOM OF BOX
292
293      Y - Y + 100
294      CALL HTY(X,Y,A)
295      WRITE(*,"E//LG"//A          ! RIGHT SIDE OF BOX
296
297      X - X - 1100
298      CALL HTY(X,Y,A)
299      WRITE(*,"E//LG"//A          ! TOP OF BOX
300      WRITE(*,"E//LE"            ! FILL THE BOX
301
302      WRITE(*,"E//MT1"            ! TEXT COLOR WHITE
303
304      X - X + 100
305      Y - Y - 75
306      CALL HTY(X,Y,A)
307      WRITE(*,"E//LP"//A          ! SET TEXT ORIGIN
308
309      IF(NOSAMP.EQ.1)THEN
310          WRITE(*,"E//LT8"//SAMPLE// "SAMPLE" ! WRITE THE TEXT
311      ELSE
312          WRITE(*,"E//LT9"//SAMPLE// "SAMPLES" ! WRITE THE TEXT
313      ENDIF
314
315      WRITE(*,"E//SC"            ! END SEGMENT
316
317      IF (CHANGE.EQ.1) GOTO 500
318
319      C _____
320      C THIS WRITE THE COLUMN HEADERS FOR " SAMPLE NAMES ", " AGENT TYPES "
321      C AND " CONCENTRATIONS "

```

# Appendix IV

```

322 C _____
323
324 ISEG - 8
325 CALL INTRPT(ISEG,SEG)
326 WRITE(*,"E//SE//SEG" ! BEGIN THE SEGMENT
327
328 WRITE(*,"E//MT4" ! TEXT COLOR BLUE
329 X - 420
330 Y - 2750
331 CALL HTY(X,Y,A)
332 WRITE(*,"E//LP//A" ! SET TEXT ORIGIN
333 TEXT - '3. SAMPLE TYPE'
334 WRITE(*,"E//LT//TEXT" ! WRITE SAMPLE TEXT
335
336 X - 1790
337 CALL HTY(X,Y,A)
338 WRITE(*,"E//LP//A" ! SET TEXT ORIGIN
339 TEXT - '4. AGENT TYPE'
340 WRITE(*,"E//LT>//TEXT" ! WRITE SAMPLE TEXT
341
342 X - 3000
343 CALL HTY(X,Y,A)
344 WRITE(*,"E//LP//A" ! SET TEXT ORIGIN
345 TEXT - '5. CONCENTRATION'
346 WRITE(*,"E//LTAX//TEXT" ! WRITE SAMPLE TEXT
347 WRITE(*,"E//SC" ! CLOSE THE SEGMENT
348
349 C _____
350 C THIS NEXT PART GETS THE SAMPLE NAME/S
351 C _____
352
353
354 67 FORMAT(SX,'IF ALL SAMPLES ARE THE SAME TYPE < Y > ',S)
355
356 68 ISAME - 0
357 INUSE - 0
358 WRITE(*,"E//LZ" ! CLEAR THE DIALOG AREA
359 WRITE(*,67)
360 READ(*,6)SAME
361
362 IF(SAME.EQ.'Y'.OR.SAME.EQ.'y')ISAME - 1
363
364 C _____
365
366 66 ISEG - 9
367 LETTER - 65 ! ITEM BEGINS WITH " A "
368 Y1 - 2600
369
370 69 X - 70
371
372 70 FORMAT(SX,'ENTER SAMPLE NAME: ',S)
373 72 FORMAT(SX,'J,A1,' ENTER SAMPLE NAME: ',S)
374
375 DO 90 I - ISTART, NOSAMP

```

# Appendix IV

```

376
377 C-----
378 C THIS WILL ONLY REQUIRE ONLY ONE INPUT IF THE SAMPLES ARE ALL THE
379 C SAME.
380 C-----
381
382 IF(SAME.EQ.1)THEN ! ALL SAMPLES ARE THE SAME
383
384 IF(INUSE.EQ.0)THEN ! ONLY READ ONCE
385 WRITE(*,*)E//LZ' ! CLEAR THE DIALOG AREA
386 75 WRITE(*,70)
387 READ(*,2,ERR=75)SAMP(I)
388
389 IF(SAMP(I).EQ.'E'.OR.SAMP(I).EQ.'e')THEN ! USER WANTS TO EXIT
390 EXIT - 1 ! SET EXIT FLAG
391 GOTO 1000 ! RETURN TO CALLER
392 ENDIF
393
394 IF(SAMP(I).EQ.'')GOTO 75 ! DO AGAIN FIELD WAS NULL
395
396 INUSE - 1 ! SET FLAG DONT COME BACK
397 GOTO 85
398 ELSE
399
400 SAMP(I) - SAMP(I - 1) ! EQUATE THE SAMPLE TYPES
401
402 GOTO 85 ! WRITE THE SAMPLE NAME
403 ENDIF
404 ENDIF
405
406
407 80 WRITE(*,*)E//LZ' ! CLEAR THE DIALOG AREA
408 WRITE(*,72)CHAR(LETTER) ! DESIGNATOR
409
410 READ(*,2,ERR=80)SAMP(I) ! INPUT SAMPLE NAME
411
412 IF(SAMP(I).EQ.'E'.OR.SAMP(I).EQ.'e')THEN ! USER WANTS TO EXIT
413 EXIT - 1 ! SET EXIT FLAG
414 GOTO 1000 ! RETURN TO CALLER
415 ENDIF
416
417 IF(SAMP(I).EQ.'')GOTO 80 ! DO AGAIN FIELD WAS NULL
418
419 85 ISEG - ISEG + 1
420
421 CALL INTRPT(ISEG,SEG)
422 WRITE(*,*)E//SK//ISEG ! DELETE THE SEGMENT
423 WRITE(*,*)E//SE//ISEG ! BEGIN THE SEGMENT
424
425 WRITE(*,*)E//MTZ' ! TEXT COLOR RED
426
427 CALL HIY(X,Y1,A)
428 WRITE(*,*)E//LP##A ! SET TEXT ORIGIN
429 TEXT - '3'//CHAR(LETTER) ! ITEM DESIGNATOR

```

# Appendix IV

```

430      WRITE(*,E//LTZ//TEXT      ! WRITE THE TEXT
431
432      WRITE(*,E//MP//          ! PANEL COLOR GRAY
433      WRITE(*,E//MT//          ! TEXT COLOR WHITE
434
435      Y - Y1 - 25
436
437      X - X + 130
438      CALL HTY(X,Y,A)
439      WRITE(*,E//LP//A          ! SET PANEL ORIGIN
440
441      X - X + 1100
442      CALL HTY(X,Y,A)
443      WRITE(*,E//LG//A          ! DRAW BOTTOM OF BOX
444
445      Y - Y + 100
446      CALL HTY(X,Y,A)
447      WRITE(*,E//LG//A          ! RIGHT SIDE OF BOX
448
449      X - X - 1100
450      CALL HTY(X,Y,A)
451      WRITE(*,E//LG//A          ! TOP OF BOX
452      WRITE(*,E//LE//          ! FILL THE BOX
453
454      Y - Y - 75
455      X - X + 50
456
457      WRITE(*,E//MT//          ! TEXT COLOR WHITE
458
459      CALL HTY(X,Y,A)
460      WRITE(*,E//LP//A          ! SET TEXT ORIGIN
461      WRITE(*,E//LTA4//SAMP(I)  ! WRITE THE TEXT
462
463      WRITE(*,E//SC//          ! CLOSE THE SEGMENT
464
465      IF(ITEM_CNG.EQ.1)GOTO 500    ! JUST CHANGING ONE ITEM
466
467      LETTER = LETTER + 1          ! INCREMENT LETTER VALUE
468      Y1 - Y1 - 125
469      X - X - 70
470
471 90  CONTINUE
472
473  C-----
474  C
475      IF (CHANGE.EQ.1) GOTO 500
476
477  C-----
478  C  THIS ASKS IF THERE IS GOING TO BE ANY CHEMICALS ADDED TO THE
479  C  SAMPLES. IF NONE ARE USED THEN I SKIP RIGHT TO THE LASER INFO.
480  C-----
481
482 100  FORMAT(5X,'WILL THERE BE AGENT ON THE SAMPLES: < Y > ',3)
483

```

# Appendix IV

```

484 110 WRITE('100)
485 READ ('6,ERR-110)ANS
486
487 WRITE('E//LZ' ! CLEAR THE DIALOG AREA
488
489 IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') GOTO 130
490 AGENT(1) = 'NONE'
491 GOTO 263 !GOTO REVIEW SCREEN
492
493 C_____
494 C THIS GETS THE TYPE OF AGENT THAT WILL BE USED ON THE SAMPLES
495 C_____
496
497 120 FORMAT(5X,'WILL AGENT TYPE BE THE SAME FOR ALL SAMPLES < Y > ',5)
498
499 130 INUSE = 0
500 ISAME = 0
501 SAME = ''
502
503 WRITE('120)
504
505 132 READ('6,ERR-132)SAME
506
507 IF(SAME.EQ.'Y'.OR.SAME.EQ.'y')ISAME = 1
508
509 C_____
510
511 WRITE('E//LZ' ! CLEAR THE DIALOG AREA
512 ISEG = 19
513 X = 1500
514 Y1 = 2600
515 LETTER = 65
516 135 FORMAT(5X,'ENTER AGENT NAME: ',5)
517 137 FORMAT(5X,'4',A1,' ENTER AGENT NAME: ',5)
518
519 138 DO 190 I = 1,ISTART, NOSAMP
520
521 C_____
522 C THIS WILL ONLY REQUIRE ONLY ONE INPUT IF THE SAMPLES ARE ALL THE
523 C SAME.
524 C_____
525
526 IF(ISAME.EQ.1)THEN ! ALL AGENTS ARE THE SAME
527
528 IF(INUSE.EQ.0)THEN ! ONLY READ ONCE
529
530 140 WRITE('135)
531 READ('2,ERR-140)AGENT(I)
532
533 IF(AGENT(I).EQ.'E'.OR.AGENT(I).EQ.'e')THEN ! USER WANTS TO EXIT
534 EXIT = 1 ! SET EXIT FLAG
535 GOTO 1000 ! RETURN TO CALLER
536 ENDF
537

```

# Appendix IV

```

538      IF(AGENT(I).EQ.'')GOTO 140      ! DO AGAIN FIELD WAS NULL
539
540      INUSE = 1                        ! SET FLAG DONT COME BACK
541
542      GOTO 160
543      ELSE
544
545      AGENT(I) = AGENT(I-1)            ! EQUATE THE AGENT TYPES
546      GOTO 160                          ! WRITE THE AGENT NAME
547      ENDIF
548      ENDIF
549
550 150  WRITE(*,137)CHAR(LETTER)          ! LIST DESIGNATOR
551
552      READ(*,2,ERR=150)AGENT(I)        ! INPUT AGENT NAME
553
554      IF(AGENT(I).EQ.'E'.OR.AGENT(I).EQ.'e')THEN ! USER WANTS TO EXIT
555      EXIT = 1                          ! SET EXIT FLAG
556      GOTO 1000                          ! RETURN TO CALLER
557      ENDIF
558
559      IF(AGENT(I).EQ.'')GOTO 150        ! DO AGAIN FIELD WAS NULL
560
561 160  ISEG = ISEG + 1
562      CALL INTRPT(ISEG,SEG)
563      WRITE(*,*)E//SK//SEG              ! DELETE THE SEGMENT
564      WRITE(*,*)E//SE//SEG              ! BEGIN THE SEGMENT
565
566      WRITE(*,*)E//MT2'                  ! TEXT COLOR RED
567
568      CALL HTY(X,Y1,A)
569      WRITE(*,*)E//LP//A                ! SET TEXT ORIGIN
570      TEXT = '4'//CHAR(LETTER)          ! ITEM DESIGNATOR
571      WRITE(*,*)E//LT2 //TEXT            ! WRITE THE TEXT
572
573      WRITE(*,*)E//MP//                  ! PANEL COLOR GRAY
574      WRITE(*,*)E//MT1'                  ! TEXT COLOR WHITE
575
576      Y = Y1 + 25
577
578      X = X + 130
579      CALL HTY(X,Y,A)
580      WRITE(*,*)E//LP//A                ! SET PANEL ORIGIN
581
582      X = X + 1100
583      CALL HTY(X,Y,A)
584      WRITE(*,*)E//LG//A                ! DRAW BOTTOM OF BOX
585
586      Y = Y + 100
587      CALL HTY(X,Y,A)
588      WRITE(*,*)E//LG//A                ! RIGHT SIDE OF BOX
589
590      X = X - 1100
591      CALL HTY(X,Y,A)

```

# Appendix IV

```

592      WRITE(*,E//LC'//A      ! TOP OF BOX
593      WRITE(*,E//LE'      ! FILL THE BOX
594
595      Y - Y - 75
596      X - X + 50
597
598      WRITE(*,E//MT1'      ! TEXT COLOR WHITE
599
600      CALL HTY(X,Y,A)
601      WRITE(*,E//LP'//A      ! SET TEXT ORIGIN
602      WRITE(*,E//LTA4'//AGENT(I)      ! WRITE THE TEXT
603
604      WRITE(*,E//SC'      ! CLOSE THE SEGMENT
605
606      IF(ITEM_CNG.EQ.1)GOTO 500      ! CHANGING ONE ITEM
607
608      LETTER - LETTER + 1      ! INCREMENT LETTER VALUE
609      Y1 - Y1 - 125
610      X - 1500
611
612 190  CONTINUE
613
614      IF (CHANGE.EQ.1) GOTO 500
615
616  C-----
617  C   THIS GETS THE AGENT CONCENTRATION OF THE AGENT FOR ALL THHE SAMPLES
618  C-----
619
620 195  FORMAT(12X,'WILL AGENT CONCENTRATION BE THE SAME FOR ALL SAMPLES')
621 198  FORMAT(35X,'< Y > ',S)
622
623 199  INUSE - 0
624      ISAME - 0
625      SAME - ''
626
627      WRITE(*,195)
628      WRITE(*,198)
629
630 200  READ(*,6,ERR=200)SAME
631      IF(SAME.EQ.'Y'.OR.SAME.EQ.'y')ISAME - 1
632
633  C-----
634
635      WRITE(*,E//LZ'      ! CLEAR THE DIALOG AREA
636      ISEG - 29
637      X - 2950
638      Y1 - 2600
639      LETTER - 65
640 210  FORMAT(5X,'ENTER CONCENTRATION: ',S)
641 212  FORMAT(5X,'S',A1,' ENTER CONCENTRATION: ',S)
642
643 215  DO 260 I - ISTART , NOSAMP
644
645  C-----

```

# Appendix IV

```

646 C THIS WILL ONLY REQUIRE ONLY ONE INPUT IF THE SAMPLES ARE ALL THE
647 C SAME.
648 C _____
649
650 IF(ISAME.EQ.1)THEN ! ALL AGENTS ARE THE SAME
651
652 IF(INUSE.EQ.0)THEN ! ONLY READ ONCE
653
654 220 WRITE(*,210)
655
656 READ(*,2,ERR=220)CONC(I)
657
658 IF(CONC(I).EQ.'E'.OR.CONC(I).EQ.'e')THEN ! USER WANTS TO EXIT
659 EXIT - 1 ! SET EXIT FLAG
660 GOTO 1000 ! RETURN TO CALLER
661 ENDIF
662
663 IF(CONC(I).EQ.'')GOTO 220 ! DO AGAIN FIELD WAS NULL
664
665 INUSE - 1 ! SET FLAG DONT COME BACK
666 GOTO 250
667 ELSE
668
669 CONC(I) - CONC(I - 1) ! EQUATE THE AGENT TYPES
670
671 GOTO 250 ! WRITE THE AGENT NAME
672 ENDIF
673 ENDIF
674
675 WRITE(*,212)CHAR(LETTER)
676
677 240 READ(*,2,ERR=240)CONC(I) ! INPUT AGENT NAME
678
679 IF(CONC(I).EQ.'E'.OR.CONC(I).EQ.'e')THEN ! USER WANTS TO EXIT
680 EXIT - 1 ! SET EXIT FLAG
681 GOTO 1000 ! RETURN TO CALLER
682 ENDIF
683
684 IF(CONC(I).EQ.'')GOTO 240 ! DO AGAIN FIELD WAS NULL
685
686 250 ISEG - ISEG + 1
687 CALL INTRPT(ISEG,SEG)
688 WRITE(*,*)E//SK//SEG ! DELETE THE SEGMENT
689 WRITE(*,*)E//SE//SEG ! BEGIN THE SEGMENT
690
691 WRITE(*,*)E//MT2 ! TEXT COLOR RED
692
693 CALL HTY(X,Y1,A)
694 WRITE(*,*)E//LP//A ! SET TEXT ORIGIN
695 TEXT - '3'//CHAR(LETTER) ! ITEM DESIGNATOR
696 WRITE(*,*)E//LT2//TEXT ! WRITE THE TEXT
697
698 WRITE(*,*)E//MP ! PANEL COLOR GRAY
699 WRITE(*,*)E//MT1 ! TEXT COLOR WHITE

```

# Appendix IV

```

700
701     Y - Y1 - 25
702
703     X - X + 130
704     CALL HTY(X,Y,A)
705     WRITE(*,*)E//LP//A           ! SET PANEL ORIGIN
706
707     X - X + 750
708     CALL HTY(X,Y,A)
709     WRITE(*,*)E//LG//A           ! DRAW BOTTOM OF BOX
710
711     Y - Y + 100
712     CALL HTY(X,Y,A)
713     WRITE(*,*)E//LG//A           ! RIGHT SIDE OF BOX
714
715     X - X - 750
716     CALL HTY(X,Y,A)
717     WRITE(*,*)E//LG//A           ! TOP OF BOX
718     WRITE(*,*)E//LE              ! FILL THE BOX
719
720     Y - Y - 75
721     X - X + 50
722
723     WRITE(*,*)E//MT1'           ! TEXT COLOR WHITE
724
725     CALL HTY(X,Y,A)
726     WRITE(*,*)E//LP//A           ! SET TEXT ORIGIN
727     WRITE(*,*)E//LT//CONC(I)     ! WRITE THE TEXT
728
729     IF(ITEM_CNG.EQ.1)GOTO 500
730
731     WRITE(*,*)E//SC              ! CLOSE THE SEGMENT
732
733     LETTER - LETTER + 1           ! INCREMENT LETTER VALUE
734     Y1 - Y1 - 125
735     X - 2950
736
737 260  CONTINUE
738
739     IF (CHANGE.EQ.1) GOTO 500
740
741  C _____
742  C   THIS DRAWS A LINE IN BLUE ACROSS THE BOTTOM OF THE SAMPLE BOXES
743  C _____
744
745 263  ISEG - 39
746     CALL INTRPT(ISEG,SEG)
747     WRITE(*,*)E//SE//SEG
748     X - 1
749     Y - Y1
750     ILINE - Y                     ! BUF THE Y FOR THE LINE
751
752     CALL HTY(X,Y,A)
753     WRITE(*,*)E//ML4'           ! LINE COLOR BLUE

```

# Appendix IV

```

754      WRITE(*,"E//LP//A"          ! SET LINE ORIGIN
755
756      X - 4095
757      CALL HTY(X,Y,A)
758      WRITE(*,"E//LG//A"          ! SET LINE END
759      WRITE(*,"E//MT4"            ! TEXT COLOR RED
760
761      X - 70
762      Y - Y - 100
763      Y2 - Y
764      CALL HTY(X,Y,A)
765      WRITE(*,"E//LP//A"          ! SET TEXT ORIGIN
766      TEXT - '%. START'
767      WRITE(*,"E//LT9//TEXT"      ! WRITE THE TEXT
768      WRITE(*,"E//FSC"            ! CLOSE THE SEGMENT
769
770      C _____
771      C   THIS PART GETS THE START, STOP, AND INCREMENT OF THE SAMPLE
772      C   STAGE FOR EACH SAMPLE. IN THIS CASE I ALLOW 1/4 DEGREE
773      C   RESOLUTION ( 500 STEPS ).
774      C _____
775
776      265  FORMAT(5X,"WILL ALL START POSITIONS BE THE SAME: < Y > ',S)
777      270  FORMAT(5X,"START POSITION OF ARM <DEGREES>: ',S)
778      272  FORMAT(5X,A2," START POSITION OF ARM <DEGREES>: ',S)
779
780      ISEG - 39                    ! SEG NUMBER BEGINS AT 40
781      Y3 - Y1
782      X - 70                      ! START POSITION OF X
783      NUM - 54                    ! ASCII VALUE OF 6
784      TPOS - 0                    ! INITIALIZE TYPE OF DAT
785
786      275  INUSE - 0
787
788      X1 - X                      ! SAVED VALUE OF START
789      Y1 - Y1 - 200               ! START POSITION OF Y
790      LETTER - 65                 ! ASCII VALUE OF A
791
792      WRITE(*,"E//L2"            ! CLEAR THE DIALOG AREA
793
794      ISAME - 0                   ! INITIALIZE DIFFERENCE
795
796      IF(TPOS.EQ.0)THEN           ! IF THIS IS START DATA
797          WRITE(*,265)             ! IN DATA FLAG
798      ELSEIF(TPOS.EQ.1)THEN       ! THIS IS END ANGLE DATA
799          WRITE(*,312)
800      ELSEIF(TPOS.EQ.2)THEN       ! THIS IS INCR ANGLE DATA
801          WRITE(*,332)
802      ENDIF
803
804      READ(*,2,ERR=280)SAME
805      IF(SAME.EQ.'Y'.OR.SAME.EQ.'y')SAME - 1
806
807      278  WRITE(*,"E//L2"            ! CLEAR THE DIALOG AREA

```

# Appendix IV

```

808
809 DO 310 I - ISTART, NOSAMP
810
811 C -----
812 C THIS WILL ONLY REQUIRE ONLY ONE INPUT IF THE SAMPLES ARE ALL THE
813 C SAME.
814 C -----
815
816 IF(I SAME.EQ.1) THEN ! ALL AGENTS ARE THE SAME
817
818 IF(INUSE.EQ.0) THEN ! ONLY READ ONCE
819 280 WRITE(*,*)E/LZ ! CLEAR DIALOG AREA
820
821 IF(IPOS.EQ.0) THEN ! IF THIS IS START DATA
822 WRITE(*,270)
823 ELSEIF(IPOS.EQ.1) THEN ! THIS IS END ANGLE DATA
824 WRITE(*,314)
825 ELSEIF(IPOS.EQ.2) THEN ! THIS IS INCR ANGLE DATA
826 WRITE(*,334)
827 ENDIF
828
829
830 READ(*,*(A5),ERR=280)POS(I)
831
832 C -----
833 C HERE I CHECK IF THE USER ADDED A DECIMAL POINT TO THE INPUT.
834 C THIS WILL BE REQUIRED FOR THE REAL INPUT. IF IT WAS NOT
835 C PROVIDED I ADD ONE. ANY VALUE IN THE 1/10th PLACE WILL BE MADE
836 C A * .5 * AS I AM LIMITING THE RESOLUTION TO 1/2 A DEGREE.
837 C -----
838
839 IDEC = 0
840 K = 0
841
842 DO 282 J = 5,1,-1 ! LOOP THRU INPUT
843
844 IF(POS(I)(J).EQ.' ')GOTO 282 ! LOOK FOR A SPACE
845
846 IF(POS(I)(J).EQ.'.')THEN ! LOOK FOR A DECIMAL
847 IDEC = IDEC + 1
848 IF(IDEC.EQ.2)GOTO 280 ! 2 DECIMALS
849
850 IF(POS(I)(J+1).GT.CHAR(48).AND. ! CHANGE ANY INPUT - .5
851 POS(I)(J+1).LT.CHAR(58))THEN ! IF NOT - .0
852 POS(I)(J+1) = '3'
853 ENDIF
854
855 GOTO 282 ! GO CONVERT TO REAL
856
857 ENDIF
858
859 IF(POS(I)(J).LT.CHAR(48).OR.POS(I)(J).GT.
860 CHAR(57))GOTO 280
861

```

# Appendix IV

```

862      IF(POS(I):J).GT.CHAR(47).AND.POS(I):J).LT.
863      CHAR(58))THEN
864
865      K = K + 1          ! COUNT NUMBERS
866
867      IF(K.EQ.4)GOTO 280    ! INPUT TOO LARGE
868      ENDIF              ! CAN ONLY BE 3 CHARS
869 282  CONTINUE
870
871      IF(K.GT.0)THEN
872      K = K + 1
873      POS(I)(K:K) = '.'    ! PLACE THE DECIMAL
874      ENDIF
875
876
877      READ(POS(I), '(P5.2)',ERR=280)RCHECK
878
879
880      WRITE(*,*)E//LZ'    ! CLEAR THE DIALOG AREA
881
882      IF(POS(I).EQ.'E'.OR.POS(I).EQ.'e')THEN ! USER WANTS TO EXIT
883      EXIT = 1            ! SET EXIT FLAG
884      GOTO 1000           ! RETURN TO CALLER
885      ENDIF
886
887      IF(POS(I).EQ.' ')GOTO 280    ! DO AGAIN FIELD WAS NULL
888      INUSE = 1             ! SET FLAG DONT COME BACK
889      GOTO 300
890      ELSE
891
892      POS(I) = POS(I - 1)    ! EQUATE THE AGENT TYPES
893
894      GOTO 300              ! WRITE THE AGENT NAME
895      ENDIF
896      ENDIF
897
898
899 285  IF(IPOS.EQ.0)THEN      ! IF THIS IS START DATA
900      WRITE(*,272)CHAR(NUM)/CHAR(LETTER)
901      ELSEIF(IPOS.EQ.1)THEN ! THIS IS END ANGLE DATA
902      WRITE(*,318)CHAR(NUM)/CHAR(LETTER)
903      ELSEIF(IPOS.EQ.2)THEN ! THIS IS INCR ANGLE DATA
904      WRITE(*,336)CHAR(NUM)/CHAR(LETTER)
905      ENDIF
906
907      POS(I) = ' '
908      K = 0
909
910 290  READ(*, '(A5)',ERR=290)POS(I)    ! INPUT AGENT NAME
911
912  C _____
913  C  HERE I CHECK IF THE USER ADDED A DECIMAL POINT TO THE INPUT.
914  C  THIS WILL BE REQUIRED FOR THE REAL INPUT. IF IT WAS NOT
915  C  PROVIDED I ADD ONE. ANY VALUE IN THE 1/10th PLACE WILL BE MADE

```

# Appendix IV

```

916 C A ".5" AS I AM LIMITING THE RESOLUTION TO 1/2 A DEGREE.
917 C-----
918
919 IDEC = 0
920 K = 0
921
922 DO 292 J = 5,1,-1 ! LOOP THRU INPUT
923
924 IF(POS(I)(J)).EQ.' ')GOTO 282 ! LOOK FOR A SPACE
925
926 IF(POS(I)(J)).EQ.'.')THEN ! LOOK FOR A DECIMAL
927 IDEC = IDEC + 1
928 IF(IDEC.EQ.2)GOTO 285 ! 2 DECIMALS
929
930 IF(POS(I)(J+1)).GT.CHAR(48).AND. ! CHANGE ANY INPUT - .5
931 POS(I)(J+1).LT.CHAR(58))THEN ! IF NOT - .0
932 POS(I)(J+1) = '5'
933 ENDIF
934
935 GOTO 297 ! GO CONVERT TO REAL
936 ENDP
937
938 IF(POS(I)(J).GT.CHAR(47).AND.POS(I)(J).LT.
939 CHAR(58))THEN
940
941 K = K + 1 ! COUNT NUMBERS
942
943 IF(K.EQ.4)GOTO 285 ! INPUT TOO LARGE
944 ENDP ! CAN ONLY BE 3 CHARS
945
946 292 CONTINUE
947
948 IF(K.GT.0)THEN
949 K = K + 1
950 POS(I)(K:K) = '.' ! PLACE THE DECIMAL
951 ENDP
952
953 READ(POS(I), '(P5.2)',ERR=295)RCHECK
954 GOTO 297
955
956 C-----
957 C THE START,STOP,AND INCREMENT OF STAGES MUST BE GOOD NUMBERS OR THE
958 C EXPERIMENT WONT WORK...
959 C-----
960
961 295 WRITE(*,'E//LZ' ! CLEAR DIALOG AREA
962 GOTO 285
963
964 C-----
965
966
967 297 IF(POS(I).EQ.'E'.OR.POS(I).EQ.'e')THEN ! USER WANTS TO EXIT
968 EXIT = 1 ! SET EXIT FLAG
969 GOTO 1000 ! RETURN TO CALLER

```

# Appendix IV

```

970      ENDIF
971
972      IF(POS(1).EQ.'')GOTO 285      ! DO AGAIN FIELD WAS NULL
973
974      300  ISEG = ISEG + 1
975      CALL INTRPT(ISEG,SEG)
976      WRITE(*,"E//SK//SEG      ! DELETE THE SEGMENT
977      WRITE(*,"E//SE//SEG      ! BEGIN THE SEGMENT
978      CALL HTY(X,Y1,A)
979      WRITE(*,"E//LP//A
980      WRITE(*,"E//MT2'      ! TEXT COLOR RED
981
982      TEXT = CHAR(NUM)//CHAR(LETTER)
983      WRITE(*,"E//LT2//TEXT      ! WRITE THE TEXT
984
985      WRITE(*,"E//MP'      ! PANEL COLOR GRAY
986      WRITE(*,"E//MT1'      ! TEXT COLOR WHITE
987
988      Y = Y1 - 25
989
990      X = X + 130
991      CALL HTY(X,Y,A)
992      WRITE(*,"E//LP//A      ! SET PANEL ORIGIN
993
994      X = X + 300
995      CALL HTY(X,Y,A)
996      WRITE(*,"E//LG//A      ! DRAW BOTTOM OF BOX
997
998      Y = Y + 100
999      CALL HTY(X,Y,A)
1000     WRITE(*,"E//LG//A      ! RIGHT SIDE OF BOX
1001
1002     X = X - 300
1003     CALL HTY(X,Y,A)
1004     WRITE(*,"E//LG//A      ! TOP OF BOX
1005     WRITE(*,"E//LE'      ! FILL THE BOX
1006
1007     Y = Y - 75
1008     X = X + 50
1009
1010     WRITE(*,"E//MT1'      ! TEXT COLOR WHITE
1011
1012     CALL HTY(X,Y,A)
1013     WRITE(*,"E//LP//A      ! SET TEXT ORIGIN
1014     WRITE(*,"E//LT5//POS(1)      ! WRITE THE TEXT
1015
1016     WRITE(*,"E//SC'      ! CLOSE THE SEGMENT
1017
1018     IF(ITEM_CNG.EQ.1)GOTO 500      ! CHANGING ONE ITEM
1019
1020     LETTER = LETTER + 1      ! INCREMENT LETTER VALUE
1021     Y1 = Y1 - 125
1022     X = X1      ! GIVE X ITS ORIGINAL
1023                  ! VALUE

```

# Appendix IV

```

1024 310 CONTINUE
1025
1026 IF(CHANGE.EQ.1)GOTO 300
1027
1028 C-----
1029
1030 IF(IPOS.EQ.0)THEN          ! THIS IS START POSITION
1031 DO 315 I = 1,NOSAMP      ! DATA.
1032   START(I) = POS(I)      ! EXCHANGE DATA
1033 315 CONTINUE
1034
1035 ELSEIF(IPOS.EQ.1)THEN      ! THIS IS STOP ANGLE
1036 DO 316 I = 1,NOSAMP      ! DATA.
1037   STOP(I) = POS(I)       ! EXCHANGE DATA
1038 316 CONTINUE
1039
1040 ELSEIF(IPOS.EQ.2)THEN      ! THIS IS INCR ANGLE
1041 DO 317 I = 1,NOSAMP      ! DATA.
1042   INC(I) = POS(I)        ! EXCHANGE DATA
1043 317 CONTINUE
1044
1045 ENDIF
1046
1047 C-----
1048 C THIS NEXT PART USES THE ABOVE SECTION OF THE ROUTINE AGAIN
1049 C BY CHANGING SOME OF THE VARIABLES, THESE ARE THE X,Y POSITIONS
1050 C OF THE SEGMENTS AD THEIR ASSIGNED NUMERIC VALUES.
1051 C THE END POSITIONS WILL BEGIN AT 50 AND THE INCREMENT SEGMENTS
1052 C WILL BEGIN AT 60.
1053 C-----
1054
1055 312 FORMAT(5X,'WILL THE END ANGLE BE THE SAME FOR ALL SAMPLES:
1056 . < Y > ',5)
1057 314 FORMAT(5X,'END POSITION OF ARM < DEGREES > : ',5)
1058 318 FORMAT(5X,A2,' END POSITION OF ARM < DEGREES > : ',5)
1059
1060 320 IPOS = IPOS + 1        ! POSITION TYPE COUNTER
1061                          ! 1 - END POSITION AND
1062                          ! 2 - INCREMENT OF ANGLE
1063 C-----
1064 C THIS IS DATA FOR THE STOP ANGLE POSITION FOR THE SAMPLES.
1065 C-----
1066
1067 IF(IPOS.EQ.1)THEN          ! THIS IS FOR END POSITION DATA
1068   ISEC = 49
1069   CALL INTRPT(ISEC,SEC)
1070   WRITE(*,*)E/PSE//SEC
1071   WRITE(*,*)E/MT4         ! TEXT COLOR BLUE
1072   X = 700
1073   Y = Y2
1074   CALL HTY(X,Y,A)
1075   WRITE(*,*)E/LP//A       ! SET TEXT ORIGIN
1076   TEXT = '7. STOP'
1077   WRITE(*,*)E/LTY//TEXT   ! WRITE THE TEXT

```

# Appendix IV

```

1078      WRITE(*,"E/SC"          ! CLOSE THE SEGMENT
1079
1080      NUM  = NUM + 1          ! INCREMENT NUMERIC VALUE
1081      X1   = X
1082      Y1   = Y - 100
1083      LETTER = 65              ! ASCII VALUE OF A
1084      ISAME = 0
1085      INUSE = 0
1086
1087 330  WRITE(*,"E/LZ"          ! CLEAR THE DIALOG AREA
1088
1089      ISAME = 0                ! INITIALIZE DIFFERENCE
1090      WRITE(*,312)             ! IN DATA FLAG
1091
1092      READ(*,2,ERR=330)ISAME
1093      IF(SAME.EQ.'Y'.OR.SAME.EQ.'y')ISAME = 1
1094      GOTO 278
1095      ENDIF
1096
1097  C-----
1098  C   THIS IS DATA FOR THE INCREMENT OF THE SAMPLES ANGLES.
1099  C-----
1100
1101      IF(POS.EQ.2)THEN          ! THIS IS FOR END POSITION DATA
1102 332  FORMAT(SX,'WILL THE INCREMENT BE THE SAME FOR ALL SAMPLES:
1103      . < Y > ',S)
1104 334  FORMAT(SX,' INCREMENT OF SAMPLE STAGE <DEGREES> : ',S)
1105 336  FORMAT(SX,A2,' INCREMENT OF SAMPLE STAGE <DEGREES> : ',S)
1106      ISEG = 99
1107      CALL INTRPT(ISEG,SEG)
1108      WRITE(*,"E/SE"/ISEG
1109      WRITE(*,"E/MT4"          ! TEXT COLOR BLUE
1110      X = 1400
1111      Y = Y2
1112      CALL HTY(X,Y,A)
1113      WRITE(*,"E/LP"/IA        ! SET TEXT ORIGIN
1114      TEXT = '8. INCR'
1115      WRITE(*,"E/LT8"/TEXT      ! WRITE THE TEXT
1116      WRITE(*,"E/SC"          ! CLOSE THE SEGMENT
1117
1118
1119      NUM  = NUM + 1          ! INCREMENT NUMERIC VALUE
1120      X1   = X
1121      Y1   = Y - 100
1122      LETTER = 65              ! ASCII VALUE OF A
1123      ISAME = 0
1124      INUSE = 0
1125
1126 340  WRITE(*,"E/LZ"          ! CLEAR THE DIALOG AREA
1127
1128      ISAME = 0                ! INITIALIZE DIFFERENCE
1129      WRITE(*,332)             ! IN DATA FLAG
1130
1131      READ(*,2,ERR=340)ISAME

```

# Appendix IV

```

1132      IF(SAME.EQ.'Y'.OR.SAME.EQ.'y')SAME = 1
1133      GOTO 278
1134      ENDIF
1135
1136 C-----
1137 C   THIS DRAWS A BLUE LINE FROM THE TEXT INPUT SCREEN TO THE BLUE LINE
1138 C   UNDER THE SAMPLE, AGENT AND CONCENTRATION SECTION. THIS AREA
1139 C   IS USED AS A TEXT INSTRUCTION AS TO HOW THE USER MIGHT MAKE
1140 C   A CORRECTION.
1141 C-----
1142
1143      ISEC = 79
1144      CALL INTRPT(ISEC,SEC)
1145      WRITE(*,*)E//SK//SEC
1146      WRITE(*,*)E//SE//SEC
1147      X = 2000
1148      Y = ILINE
1149      CALL HTY(X,Y,A)
1150      WRITE(*,*)E//MLA'          ! LINE COLOR BLUE
1151      WRITE(*,*)E//LP//A        ! SET LINE ORIGIN
1152      Y = 450
1153      CALL HTY(X,Y,A)
1154      WRITE(*,*)E//LG//A        ! SET LINE END
1155      X = X + 20
1156      Y = ILINE - 20
1157      CALL HTY(X,Y,A)
1158      WRITE(*,*)E//LP//A        ! SET LINE ORIGIN
1159      Y = 450
1160      CALL HTY(X,Y,A)
1161      WRITE(*,*)E//LG//A        ! SET LINE END
1162      Y = ILINE - 20
1163      CALL HTY(X,Y,A)
1164      WRITE(*,*)E//LP//A        ! SET LINE ORIGIN
1165      X = 4075
1166      CALL HTY(X,Y,A)
1167      WRITE(*,*)E//LG//A        ! SET LINE END
1168      X = 4095
1169      Y = ILINE
1170      CALL HTY(X,Y,A)
1171      WRITE(*,*)E//LP//A        ! SET LINE ORIGIN
1172      Y = 450
1173      CALL HTY(X,Y,A)
1174      WRITE(*,*)E//LG//A        ! SET LINE END
1175      Y = ILINE - 20
1176      X = X - 20
1177      CALL HTY(X,Y,A)
1178      WRITE(*,*)E//LP//A        ! SET LINE ORIGIN
1179      Y = 450
1180      CALL HTY(X,Y,A)
1181      WRITE(*,*)E//LG//A        ! SET LINE END
1182
1183 C-----
1184 C   THIS PLACES TEXT IN THE PANEL THAT WAS JUST OUTLINED ABOVE.
1185 C   TEXT COLOR IS RED.

```

# Appendix IV

```

1186 C-----
1187
1188 TEXT - "***** CORRECTIONS *****"
1189
1190 X - 2350
1191 Y - ILINE - 200
1192 CALL HTY(X,Y,A)
1193 WRITE(*,"E//LP//A
1194 WRITE(*,"E//MTX"
1195 WRITE(*,"E//LTA<//TEXT
1196
1197 TEXT - "Enter a single number to"
1198 Y - Y - 200
1199 X1 - X - 150
1200
1201 CALL HTY(X,Y,A)
1202 WRITE(*,"E//LP//A
1203 WRITE(*,"E//MTX"
1204 WRITE(*,"E//LTA<//TEXT
1205
1206 TEXT - "change a whole group."
1207 Y - Y - 100
1208 CALL HTY(X,Y,A)
1209 WRITE(*,"E//LP//A
1210 WRITE(*,"E//LTA<//TEXT
1211
1212
1213 TEXT - "Enter a single number with"
1214 Y - Y - 200
1215 CALL HTY(X,Y,A)
1216 WRITE(*,"E//LP//A
1217 WRITE(*,"E//LTA<//TEXT
1218
1219
1220 TEXT - "a letter to change a specific"
1221 Y - Y - 100
1222 CALL HTY(X,Y,A)
1223 WRITE(*,"E//LP//A
1224 WRITE(*,"E//LTA<//TEXT
1225
1226 TEXT - "entry."
1227 Y - Y - 100
1228 CALL HTY(X,Y,A)
1229 WRITE(*,"E//LP//A
1230 WRITE(*,"E//LTA<//TEXT
1231
1232
1233 TEXT - "PRESS RETURN TO CONTINUE"
1234 Y - Y - 200
1235 X - X1
1236 CALL HTY(X,Y,A)
1237 WRITE(*,"E//LP//A
1238 WRITE(*,"E//MTX"
1239 WRITE(*,"E//LTA<//TEXT

```

# Appendix IV

```

1240
1241     WRITE(*,*)E//SC'          ! END THE SEGMENT
1242
1243 C-----
1244 C   THIS PART ALLOWS THE USER TO MAKE CORRECTIONS TO ANY DATA ITEM
1245 C   BEFORE MOVING FORWARD.
1246 C-----
1247
1248 500  CHANGE  - 0              ! INITIALIZE CORRECTION FLAGS
1249     ITEM_CNG - 0
1250     ISTART  - 1
1251
1252 345  WRITE(*,*)E//LZ'        ! CLEAR THE SCREEN
1253
1254     CORRECT - ''              ! INITIALIZE THE ERROR INPUT
1255
1256
1257 350  FORMAT(SX,'ENTER A NUMBER FOR CORRECTIONS: ',S)
1258
1259
1260     WRITE(*,350)
1261     READ(*, '(A2)',ERR=345)CORRECT
1262
1263
1264     IF(CORRECT.EQ.' ')GOTO 1000    ! USER WANTS TO GO ON
1265
1266 C-----
1267 C   THIS EVALUATES THE FIRST CHARACTER TO MAKE SURE IT IS A NUMBER
1268 C-----
1269
1270     IF(ICHAR(CORRECT(1:1)).GT.48.AND.
1271        ICHAR(CORRECT(1:1)).LT.58)GOTO 360
1272     GOTO 345          ! FIRST CHARACTER WAS NOT A NUMBER
1273
1274 C-----
1275 C   THIS CHECKS THE SECOND CHARACTER TO SEE IF IT IS THERE OR THAT
1276 C   IT IS A CHARACTER FROM A - H DEPENDING ON THE NUMBER OF SAMPLES.
1277 C-----
1278
1279 360  IF(CORRECT(2:2).EQ.' ')THEN    ! USER WANTS TO CHANGE AN
1280        ! ENTIRE GROUP
1281     READ(CORRECT(1:1), '(BN,I2)',ERR=345)NEW_INPUT
1282
1283     IF(NEW_INPUT.LT.1.OR.NEW_INPUT.GT.8)GOTO 345
1284
1285     CHANGE - 1
1286     IF(NEW_INPUT.LT.6)THEN          ! GO CHANGE SAMPLE DATA
1287        GOTO(40,60,80,110,199)NEW_INPUT  ! ITEMS 1 - 5
1288
1289     ELSE                            ! STAGE MOVEMENT DATA
1290
1291     IF(NEW_INPUT.EQ.6)THEN
1292        IPO5 - 0
1293        ISEC - 39

```

# Appendix IV

```

1294      X - 70          ! START POSITION OF X
1295      NUM - 54        ! ASCII VALUE OF 6
1296
1297      ELSEIF(NEW_INPUT.EQ.7)THEN
1298      IPOS - 1
1299      ISEC - 49
1300      X - 700          ! START POSITION OF X
1301      NUM - 55         ! ASCII VALUE OF 7
1302
1303
1304      ELSEIF(NEW_INPUT.EQ.8)THEN
1305      IPOS - 2
1306      ISEC - 99
1307      X - 1400         ! START POSITION OF X
1308      NUM - 56         ! ASCII VALUE OF 8
1309
1310      ENDIF
1311      Y1 - Y3
1312
1313      GOTO 275
1314
1315      FNDIF
1316      ENDIF
1317
1318      C-----
1319      C  THIS PART IS ENTERED IF THE USER IS SELECTING A SPECIFIC ITEM
1320      C  ON THE MENU TO CHANGE. FIRST I GET THE INTEGER AND COMPARE IT
1321      C  TO THE NUMBER OF SAMPLES. A BAD ENTRY MEANS DO IT AGAIN.
1322      C  THEN I TAKE THE SECOND ENTRY AND MAKE SURE ITS CORRECT AND
1323      C  WITHIN BOUNDS.
1324      C-----
1325
1326      READ(CORRECT(1:1),'(BN,D)',ERR=345)NEW_INPUT ! CNG TO INTEG
1327
1328      IF(NEW_INPUT.LT.1.OR.NEW_INPUT.GT.8)GOTO 345 ! BAD INPUT
1329
1330      LET - ICHAR(CORRECT(2:2))          ! CNG TO INTEGER
1331
1332      C-----
1333      C  THIS CHECKS FOR CASE. IF INPUT WAS LOWER CASE I MAKE IT UPPER
1334      C  CASE HERE. a - h is changed to upper case
1335      C-----
1336
1337      IF(LET.GT.96.AND.LET.LT.105)THEN    ! ITS LOWER CASE
1338      LET - LET - 32                      ! MAKE LOWER CASE
1339      ENDIF
1340
1341      IF(LET.LT.65.OR.LET.GT.64 + NOSAMP)GOTO 345 ! BAD INPUT
1342
1343      C-----
1344      C  HERE I SET GET THE SPECIFIC SEGMENT NUMBER I NEED AND
1345      C  THEN I SET ISTART FOR THE CORRECT ARRAY VALUE. I AM A SINGLE
1346      C  EVENT DATA FLAG * ITEM_CNG - 1 * AND I GO GET THE DATA.
1347      C-----

```

# Appendix IV

```

1348
1349      ISTART - LET - 64      ! GET ITEM NUMBER
1350      ISAME - 0      ! FLAG FOR SINGLE INPUT
1351      ITEM_CNG - 1      ! FLAG TO JUMP OUT
1352
1353      LETTER - LET
1354
1355      IF(NEW_INPUT.EQ.3)THEN      ! CORRECTING A SAMPLE ITEM
1356          ISEG - 8 + ISTART      ! THIS IS THE SEGMENT No.
1357          Y1 - 2600 - (125 * (ISTART - 1))      ! Y VECTOR OF SEGMENT
1358          GOTO 69
1359
1360      C_____
1361      C   AGENT DATA
1362      C_____
1363
1364      ELSEIF(NEW_INPUT.EQ.4)THEN      ! CORRECTING AN AGENT ITEM
1365          IF(AGENT(1).EQ.'NONE')GOTO 345      ! CANT PICK SPECIFIC
1366              ! NOTHING.
1367
1368          ISEG - 18 + ISTART      ! THIS IS THE SEGMENT No.
1369          Y1 - 2600 - (125 * (ISTART - 1))      ! Y VECTOR OF SEGMENT
1370          X - 1500
1371          GOTO 138
1372
1373
1374      C_____
1375      C   CONCENTRATION DATA
1376      C_____
1377
1378      ELSEIF(NEW_INPUT.EQ.5)THEN      ! CORRECTING AN AGENT ITEM
1379
1380          IF(AGENT(1).EQ.'NONE')GOTO 345      ! CANT PICK SPECIFIC
1381              ! NOTHING.
1382
1383          ISEG - 28 + ISTART      ! THIS IS THE SEGMENT No.
1384          Y1 - 2600 - (125 * (ISTART - 1))      ! Y VECTOR OF SEGMENT
1385          X - 2950
1386          GOTO 215
1387
1388      C_____
1389      C   SAMPLE STAGE START POSITION
1390      C_____
1391
1392      ELSEIF(NEW_INPUT.EQ.6)THEN      ! CORRECTING START POS
1393          IFOS - 0
1394          ISEG - 38 + ISTART      ! THIS IS THE SEGMENT No.
1395
1396          Y1 - Y3 - (125 * (ISTART - 1))      ! Y VECTOR OF SEGMENT
1397          Y1 - Y1 - 200
1398          X - 70
1399          NUM - 54      ! ASCII VALUE OF 6
1400          GOTO 278
1401

```

# Appendix IV

```

1402 C _____
1403 C   SAMPLE STAGE STOP POSITION
1404 C _____
1405
1406 ELSEIF(NEW_INPUT.EQ.7)THEN      ! CORRECTING STOP POS
1407     IPOS - 1
1408     ISEG - 48 + ISTART           ! THIS IS THE SEGMENT No.
1409     Y1 - Y3 - (125 * (ISTART - 1)) ! Y VECTOR OF SEGMENT
1410     X - 700
1411     Y1 - Y1 - 200
1412     NUM - 55                    ! ASCII VALUE OF 7
1413     GOTO 278
1414
1415 C _____
1416 C   SAMPLE STAGE INCREMENT
1417 C _____
1418
1419 ELSEIF(NEW_INPUT.EQ.8)THEN      ! CORRECTING INCREMENT
1420     IPOS - 2
1421     ISEG - 58 + ISTART           ! THIS IS THE SEGMENT No.
1422     Y1 - Y3 - (125 * (ISTART - 1)) ! Y VECTOR OF SEGMENT
1423     X - 1400
1424     Y1 - Y1 - 200
1425     NUM - 56                    ! ASCII VALUE OF 8
1426     GOTO 278
1427
1428
1429     ENDIF
1430
1431 1000 RETURN
1432     END

```

## Appendix IV

### *AIV.25 Analog APSD Software Modules: TEK\_TEXT Source Code.*

```

1      SUBROUTINE TEK_TEXT(TXT_FLG,PORT,RESET,ANS,ANALYTE,
2          AMOUNT,IEXIT)
3
4      C-----
5      C  THIS ROUTINE IS CALLED JUST TO WRITE GRAPHIC TEXT FOR THE USER
6      C
7      C  TXT_FLG - 1
8      C  FIRST THE PORT SETTINGS ARE DEFINED AND THE USER IS REQUESTED
9      C  TO ENTER THE PORT UPON WHICH ALL COMMUNICATIONS WILL TAKE PLACE.
10     C
11     C  TXT_FLG - 2
12     C  HERE THE USER IS ASKED IF THE REAL TIME GRAPHICS WILL BEUSED
13     C  OR A/D VOLTAGES DISPLAYED OR IF NO OUTPUT AT ALL IS DESIRED.
14     C
15     C  TXT_FLG - 3
16     C  THIS PART IS USED TO TELL THE USER TO APPLY AGENT TO THE
17     C  SAMPLE AND HOW MUCH.
18     C
19     C  TXT_FLG - 4
20     C  THIS TELLS THE USER THAT A SPECIFIC LASER IS READY TO BE
21     C  CALCULATED.
22     C
23     C  PORT - THE SERIAL COMMUNICATIONS PORT SUPPLIED BY THE USER
24     C  SENT BACK TO THE CALLER
25     C
26     C  RESET - 0      ! SENT FROM THE CALLER
27     C  DRAW THE GRAPHICS PANEL FOR TEXT
28     C
29     C  RESET - 1
30     C  DONT DO ANYTHING TO THE PANEL DRAW/DELETE
31     C
32     C  RESET - 2
33     C  DELETE THE PANEL AND TEXT
34     C
35     C  ANS - CHARACTER NUMBER 1,2 OR 3. FOR THE TYPE OF OUTPUT THE
36     C  USER WOULD LIKE TO HAVE FOR THIS EXPERIMENT.
37     C  SENT BACK TO THE CALLER
38     C
39     C  ANALYTE - THE CHEMICAL THAT IS TO BE APPLIED TO THE CURRENT
40     C  SAMPLE. SENT IN FROM CALLER.
41     C
42     C  AMOUNT - THE AMOUNT OF THE CHEMICAL THAT IS TO BE USED ON THE
43     C  CURRENT SAMPLE. SENT IN FROM THE CALLER
44     C  IEXIT - FLAG THAT USER WANTS TO EXIT THE PROGRAM
45     C-----
46
47     CHARACTER E,SEG*3,A*5,A1*5,TEXT*80,PORT*10,ANS*1,FINISHED*1
48     CHARACTER ANALYTE*20,AMOUNT*15
49
50     INTEGER X,Y,TXT_FLG,RESET
51

```

# Appendix IV

```

52      E - CHAR(27)
53
54
55      WRITE(*,*)E//LV0
56      WRITE(*,*)E//LZ
57
58
59      ISEG - 900
60      CALL INTRPT(ISEG,SEG)
61
62
63      IF(RESET.EQ.2)THEN          ! CLEAR THE WHOLE VIEW
64
65          WRITE(*,*)E//SK//SEG    ! DELETE SEGMENT 900
66          ISEG - 901
67          CALL INTRPT(ISEG,SEG)
68          WRITE(*,*)E//SK//SEG    ! DELETE SEGMENT 900
69          WRITE(*,*)E//KN0
70          GOTO 1010
71
72      ELSEIF(RESET.EQ.0)THEN
73
74          WRITE(*,*)E//SE//SEG    ! BEGIN SEGMENT 900
75          ! USER WANTS A RED PANEL
76          WRITE(*,*)E//MP
77          WRITE(*,*)E//MT
78          WRITE(*,*)E//ML1
79          X - 500
80          Y - 2800
81          CALL HTY(X,Y,A)
82          WRITE(*,*)E//LP//A//Y
83
84          X - X + 3095
85          CALL HTY(X,Y,A)
86          WRITE(*,*)E//LG//A
87
88          Y - Y - 2000
89          CALL HTY(X,Y,A)
90          WRITE(*,*)E//LG//A
91
92          X - X - 3095
93          CALL HTY(X,Y,A)
94          WRITE(*,*)E//LG//A
95
96          WRITE(*,*)E//SC
97
98      ENDIF
99
100  C-----
101  C   THIS DRAWS THE TEXT IN THE PANEL
102  C-----
103
104      ISEG - 901
105

```

# Appendix IV

```

106      CALL INTRPT(ISEG,SEG)
107      WRITE(*,"E/PSK"/SEG      ! DELETE SEGMENT 901
108      WRITE(*,"E/PKN0"        ! RENEW THE VIEW
109      WRITE(*,"E/SE"/SEG      ! BEGIN SEGMENT 901
110
111      IF(TXT_FLG.EQ.1)THEN      ! SET PORT PARAMETERS
112
113      X - 1000
114      Y - 2500
115      CALL HTY(X,Y,A)
116      WRITE(*,"E/LP"/A
117      TEXT - 'THE COMMUNICATIONS PORT MUST BE DEFINED WITH'
118      WRITE(*,"E/LTD0"/TEXT
119      Y - Y - 150
120      CALL HTY(X,Y,A)
121      WRITE(*,"E/LP"/A
122
123      TEXT - 'THE FOLLOWING PARAMETERS:'
124      WRITE(*,"E/LTD0"/TEXT
125
126      X - X + 200
127      Y - Y - 300
128      CALL HTY(X,Y,A)
129      WRITE(*,"E/LP"/A
130
131      TEXT - '9600 BAUD      NO PARITY'
132      WRITE(*,"E/LTD0"/TEXT
133      Y - Y - 150
134      CALL HTY(X,Y,A)
135      WRITE(*,"E/LP"/A
136
137      TEXT - '8  BITS      1 STOP BIT'
138      WRITE(*,"E/LTD0"/TEXT
139
140      Y - Y - 150
141      CALL HTY(X,Y,A)
142      WRITE(*,"E/LP"/A
143      TEXT - 'NO ECHO      NO LOCAL ECHO'
144      WRITE(*,"E/LTD0"/TEXT
145
146      Y - Y - 150
147      CALL HTY(X,Y,A)
148      WRITE(*,"E/LP"/A
149      TEXT - 'PASSALL MODE '
150      WRITE(*,"E/LTD0"/TEXT
151
152      Y - Y - 300
153      CALL HTY(X,Y,A)
154      WRITE(*,"E/LP"/A
155      TEXT - 'PRESS RETURN FOR DEFAULT PORT < TXA2 >'
156      WRITE(*,"E/LTD0"/TEXT
157
158      Y - Y - 350
159      CALL HTY(X,Y,A)

```

# Appendix IV

```

160      WRITE(*,"E//LP//A
161      TEXT - 'ENTER THE SERIAL PORT NAME: '
162      WRITE(*,"E//LTD0//TEXT
163
164      WRITE(*,"E//SC"
165
166      WRITE(*,"E//LTZ"      ! TEXT YELLOW ON RED
167      WRITE(*,"E//LLZ"      ! DIALOG AREA 2 LINES
168      WRITE(*,"E//LC"       ! DIALOG AREA 10 CHARACTERS LONG
169
170      X = 2650
171      Y = Y + 90
172      CALL HTY(X,Y,A)
173      WRITE(*,"E//LX//A
174
175      WRITE(*,"E//LV1"
176 50    WRITE(*,"E//LZ"
177      READ(*,"(A5),ERR = 50)PORT
178
179      C-----
180      C    CHECK FOR PORT DEFINITION ERRORS
181      C-----
182
183      IF(PORT(1:1).EQ.CHAR(116))PORT(1:1) = 'T'
184      IF(PORT(2:2).EQ.CHAR(120))PORT(2:2) = 'X'
185      IF(PORT(3:3).EQ.CHAR(97))PORT(3:3) = 'A'
186
187      IF(PORT.EQ.'')THEN      ! USER WANTS DEFAULT
188      PORT = 'TXA2'          ! SERIAL PORT
189      GOTO 1000
190      ENDIF
191
192      IF(PORT(1:3).EQ.TXA)THEN ! THESE ARE THE SERIAL
193      ! PORTS IM USING
194
195      DO 60 I = 1,8          ! LOOP THRU 0 - 7
196      IF(PORT(4:4).EQ.CHAR(47 + I))GOTO 70
197 60    CONTINUE
198
199      GOTO 50                ! INPUT WAS BAD DO IT AGAIN
200 70    PORT(3:3) = ':'      ! MAKE SURE YOU END WITH A COLON
201
202      ELSE
203      GOTO 50                ! BAD INPUT
204      ENDIF
205
206      C-----
207      C    HERE THE USER IS ASKED TO SELECT THE TYPE OF OUT PUT DESIRED
208      C-----
209
210      ELSEIF(TXT_PLG.EQ.2)THEN
211
212      X = 1000
213      Y = 2500

```

# Appendix IV

```

214     CALL HTY(X,Y,A)
215     WRITE(*,"E//LP//A
216     TEXT - 'ENTER THE TYPE OF OUTPUT DESIRED:'
217     WRITE(*,"E//LTD0//TEXT
218
219     Y - Y - 400
220     CALL HTY(X,Y,A)
221     WRITE(*,"E//LP//A
222     TEXT - '1. REAL TIME TEKTRONIX GRAPHICS DISPLAY'
223     WRITE(*,"E//LTD0//TEXT
224
225     Y - Y - 150
226     CALL HTY(X,Y,A)
227     WRITE(*,"E//LP//A
228     TEXT - '2. REAL TIME A/D CHANNEL VOLTAGE OUTPUTS'
229     WRITE(*,"E//LTD0//TEXT
230
231
232     Y - Y - 150
233     CALL HTY(X,Y,A)
234     WRITE(*,"E//LP//A
235     TEXT - '3. NO DISPLAY OF DATA'
236     WRITE(*,"E//LTD0//TEXT
237
238     WRITE(*,"E//SC"      ! END THE SEGMENT
239
240     WRITE(*,"E//LT72"    ! TEXT YELLOW ON RED
241     WRITE(*,"E//LL2"     ! DIALOG AREA 2 LINES
242     WRITE(*,"E//LC:"     ! DIALOG AREA 10 CHARACTERS LONG
243
244     X - 2000
245     Y - Y - 300
246     CALL HTY(X,Y,A)
247     WRITE(*,"E//LX//A
248
249     WRITE(*,"E//LV1'
250 80  WRITE(*,"E//LZ'
251
252     READ*,"(A),ERR=80)ANS
253
254     IF(ANS.EQ.'1'.OR.ANS.EQ.'2'.OR.ANS.EQ.'3')GOTO 1000
255     GOTO 80
256
257 C-----
258 C   THIS TELLS THE USER TO PUT THE ANALYTE ON THE SAMPLE. THE SAMPLE
259 C   HAS BEEN MOVED TO THE CORRECT POSITION TO ADD THE CHEMICAL
260 C-----
261
262     ELSEIF(TXT_PLG.EQ.3)THEN
263
264     X - 1000
265     Y - 2000
266     CALL HTY(X,Y,A)
267     WRITE(*,"E//LP//A

```

# Appendix IV

```

268      TEXT - 'THE SAMPLE IS NOW READY FOR THE '//ANALYTE
269
270      WRITEC,"E//LTD0//TEXT
271
272      Y - Y - 200
273      CALL HTY(X,Y,A)
274      WRITEC,"E//LP//A
275      TEXT - 'PLEASE APPLY: '//AMOUNT//' TO THE SAMPLE'
276      WRITEC,"E//LTD0//TEXT
277
278      X - X + 650
279      Y - Y - 300
280      CALL HTY(X,Y,A)
281      WRITEC,"E//LP//A
282      TEXT - 'PRESS TO GO ON.'
283      WRITEC,"E//LTD0//TEXT
284
285      WRITEC,"E//SC          ! END THE SEGMENT
286
287      READC,"(A),ERR-1000)FINISHED
288      READC,"(A),ERR-1000)FINISHED
289
290      C-----
291
292      ELSEIF(TXT_FLC.EQ.4)THEN
293
294      X - 1200
295      Y - 2300
296      CALL HTY(X,Y,A)
297      WRITEC,"E//LP//A
298
299      TEXT - 'TUNE LASER No. '//ANS//' TO WAVELENGTH: '//ANALYTE
300      WRITEC,"E//LTD0//TEXT
301
302      X - 850
303      Y - Y - 800
304      CALL HTY(X,Y,A)
305      WRITEC,"E//LP//A
306      TEXT - 'PRESS RETURN WHEN THE CALIBRATION IS COMPLETE'
307      WRITEC,"E//LTD0//TEXT
308
309      WRITEC,"E//SC          ! END THE SEGMENT
310
311      READC,"(A),ERR-1000)FINISHED
312
313      ELSE
314      EXIT - 0
315      ENDIF
316
317      1000  WRITEC,"E//LV0          ! DISABLE DIALOG AREA
318      WRITEC,"E//LZ          ! CLEAR DIALOG AREA
319      WRITEC,"E//LN144          ! TEXT WHITE ON BLUE
320      WRITEC,"E//LL2          ! DIALOG AREA 2 LINES
321      X - 0

```

## Appendix IV

```
322      Y = 0
323      CALL HTY(X,Y,A)
324      WRITE(*,E//LX'//A      ! RESET DIALOG AREA POSITION
325      WRITE(*,E//LCB'      ! DIALOG AREA 80 CHARACTERS LONG
326
327 1010  RESET = 0      ! RESET THE PANEL DEL/DRAW FLC
328      RETURN
329
330      END
```

## Appendix IV

### AIV.26 Analog APSD Software Modules: TERM\_INFO Source Code.

```

1      SUBROUTINE TERM_INFO(ITERM,NUM_PLANES,TOT_MEM,FREE_MEM,
2      . IVERSION,OPT_NUM,OPT_INFO)
3
4      C-----
5      C THIS ROUTINE WILL POLL THE TERMINAL THAT YOU ARE LOGGED IN ON TO SEE
6      C WHAT TEKTRONIX MODEL # IT IS (AND IF IT IS A TEK TERMINAL AT ALL), AND
7      C ALSO WILL TRY TO DETERMINE HOW MANY BIT PLANES THE TERMINAL HAS
8      C INSTALLED,
9      C THE TOTAL MEMORY INSTALLED, THE AVAILABLE MEMORY, AND THE FIRMWARE
10     C VERSION
11     C NUMBER FOR THE TERMINAL (AND ALSO THE FIRMWARE VERSION NUMBER FOR ONE
12     C ADDITIONAL OPTION, IF REQUESTED).
13     C-----
14     C PARAMETERS:
15     C
16     C IUT      - THE LOGICAL UNIT # OF THE TERMINAL FOR I/O PURPOSES
17     C ITERM    - THE INTEGER CONTAINING THE TERM TYPE (I.E. '4125')
18     C NUM_PLANES - THE ACTUAL # OF BIT PLANES IN THE TERMINAL. (RETURNS
19     C      NUM_PLANES--1 IF CANNOT DETERMINE THE # PRESENT)
20     C TOT_MEM  - TOTAL AMOUNT OF MEMORY INSTALLED IN THE TERMINAL(K BYTES)
21     C FREE_MEM - MEMORY PRESENTLY AVAILABLE FOR USE (K BYTES)
22     C IVERSION - VERSION NUMBER OF THE STANDARD FIRMWARE IN THE TERMINAL
23     C OPT_NUM  - AN OPTIONAL FIRMWARE VERSION NUMBER REQUEST(0=NO REQUEST)
24     C OPT_INFO - THE RETURNED VALUE FROM 'OPT_NUM' CALL ABOVE
25     C-----
26     C
27     CHARACTER E*1,VERSION*3,OPT_NUM*2,OPT_INFO*3
28     CHARACTER A*20,A1*2,ATERM*3,B*1
29     INTEGER TOT_MEM,FREE_MEM,OPT22,OPT23
30     C
31     C-----
32     C
33     E=CHAR(27)      ! ESCAPE CHARACTER
34
35     C THE FOLLOWING WILL DETERMINE IF THE TERM IS A TEK GRAPHICS TERMINAL
36
37     WRITE(*,"E//%0"  ! PUT TERMINAL INTO TEK MODE
38     WRITE(*,"E//LV"  ! MAKE DIALOG AREA INVISIBLE
39     WRITE(*,"E//IQ?"  ! GET THE TERMINAL TYPE
40
41     5  READ(*,1,ERR=5)A
42
43     IF(A.EQ."")GOTO 100      ! USER HIT <CR>, SO NOT A TEK TERM.
44     IF(A(1:2).NE."7")GOTO 100 ! INCORRECT TERMINAL RESPONSE
45
46     IADD = ICHAR(A(3:3)) + 32
47     IADD1 = ICHAR(A(4:4)) + 32
48
49     ATERM=A(3:5)
50
51     CALL DECODE(ATERM,ITERM) ! CONVERT TEK CHAR CODE TO INTEGER

```

# Appendix IV

```

52
53
54
55     IF(ITEM.EQ.4111)THEN          ! -- FOR 4111 TERMINAL ONLY --
56
57         WRITE(*,"E//1Q7P"      ! TOTAL MEMORY(# OF 16 BYTE BLOCKS)
58 10    READ(*,1,ERR=10)A         ! (4111 OPT 2C, PAGED MEMORY)
59
60
61         ATERM=A(3:5)
62
63         CALL DECODE(ATERM,TOT_MEM) ! CONVERT TEK CHAR CODE TO INTEGER
64
65
66         ATERM=A(6:8)
67         CALL DECODE(ATERM,FREE_MEM) ! CONVERT TEK CHAR CODE TO INTEGER
68
69     ELSE                          ! -- ALL OTHER TERMINALS --
70
71         WRITE(*,"E//1Q7M"
72
73 20    READ(*,1,ERR=20)A
74
75         ATERM=A(3:5)
76
77         CALL DECODE(ATERM,TOT_MEM) ! CONVERT TEK CHAR CODE TO INTEGER
78
79         ATERM=A(6:8)
80         CALL DECODE(ATERM,FREE_MEM) ! CONVERT TEK CHAR CODE TO INTEGER
81
82         ! IN KB
83     END IF
84
85     WRITE(*,"E//1Q00"          ! GET THE STD FIRMWARE VERSION
86     READ(*,1,ERR=30)A
87
88 30    VERSION=A(3:5)
89
90     CALL DECODE(VERSION,IVERSION) ! CONVERT TEK CHAR CODE TO INTEGER
91
92
93     IF(ITEM.EQ.4115.OR.ITEM.EQ.4129.OR.ITEM.EQ.4128
94     .OR.ITEM.EQ.4225.OR.ITEM.EQ.4236)THEN
95
96         WRITE(*,"E//1Q22"      ! GET THE OPT22 INFO
97 40    READ(*,1,ERR=40)A
98
99         CALL DECODE(A(3:5),OPT22) ! CONVERT TEK CHAR CODE TO INTEGER
100
101        WRITE(*,"E//1Q23"      ! GET THE OPT23 INFO
102 50    READ(*,1,ERR=50)A
103
104        CALL DECODE(A(3:5),OPT23) ! CONVERT TEK CHAR CODE TO INTEGER
105

```

# Appendix IV

```

106      IF(OPT22.EQ.0.AND.OPT23.EQ.1)THEN
107          NUM_PLANES=8
108      ELSE
109          NUM_PLANES=4
110      END IF
111
112      ELSE
113          NUM_PLANES=4
114      END IF
115
116      GOTO 1000          ! DONE, SO EXIT
117
118 100  CONTINUE          ! NOT A TEK TERMINAL, SO EXIT
119
120      ITERM = 0
121      NUM_PLANES = -1
122
123  C-----
124
125  1  FORMAT(A20)
126
127 1000 CONTINUE
128      RETURN
129      END

```

## Appendix IV

### *AIV.27 Analog APSD Software Modules: TIME4 Source Code.*

```
1      SUBROUTINE TIME4(DBUFF,TBUFF)
2
3      C _____
4      C SUBROUTINE TO GET AN ASCII STRING WITH THE DATE AND TIME
5      C _____
6
7      CHARACTER*9 DBUFF
8      CHARACTER*8 TBUFF
9
10     CALL DATE(DBUFF)
11     CALL TIME(TBUFF)
12
13     RETURN
14     END
```

# Appendix IV

## AIV.28 Analog APSD Software Modules: TWAIT Source Code.

```

1      SUBROUTINE TWAIT(TWTIM)
2      C _____
3      C SUBROUTINE TO WAIT FOR 'TWTIM' TENTHS OF A SECOND AND THEN RETURN
4      C _____
5      C ARNE A. JOHNSON, 21 APRIL 88
6      C _____
7      REAL TIME1,FWTIM,TMIN
8      MM=1
9      IFLAG=0
10     C _____
11     C TIME DELAY LOOP (FOR 'TWTIM' TENTHS OF A SECOND)
12     C
13     IF(TWTIM.EQ.0)GOTO 1000
14
15
16     100 CONTINUE
17     IFLAG=IFLAG+1
18     TIME1=SECNDS(0.0)
19     FWTIM=FLOAT(TWTIM)
20     TMIN=FWTIM/10.
21
22     50 CONTINUE
23
24     DO 55 M=1,20      ! MEANINGLESS CALC. TO TAKE UP TIME
25     MM=M*10+MM
26     55 CONTINUE
27
28     DELTA=SECNDS(TIME1)
29     C WRITE(*,*)DELTA
30     IF (DELTA.LT.TMIN) GO TO 50
31     C CLOSE(1)
32
33     1000 RETURN
34     END
35
36
37     $
38     SUBROUTINE TWAIT1(IWTIM)
39     C _____
40     C SUBROUTINE TO WAIT FOR 'TWTIM' TENTHS OF A SECOND AND THEN RETURN
41     C _____
42     REAL TIME1,FWTIM,TMIN
43     MM=1
44     IFLAG=0
45     C _____
46     C TIME DELAY LOOP (FOR 'TWTIM' TENTHS OF A SECOND)
47     C
48     IF(TWTIM.EQ.0)GOTO 1000
49
50
51     100 CONTINUE

```

# Appendix IV

```
52      IFLAG=IFLAG+1
53      TIME1=SECNDS(0.0)
54      FWTIM=FLOAT(TWTIM)
55      TMIN=FWTIM/10.
56
57  50  CONTINUE
58
59      DO 55 M=1,20      ! MEANINGLESS CALC. TO TAKE UP TIME
60          MM=M*10+MM
61  55  CONTINUE
62
63      DELTA=SECNDS(TIME1)
64  C   WRITE(*,*)DELTA
65      IF (DELTA.LT.TMIN) GO TO 50
66  C   CLOSE(1)
67
68  1000 RETURN
69      END
```

# Appendix IV

## AIV.29 Analog APSD Software Modules: V45 Source Code.

```

1      SUBROUTINE V45(SHOTS,INUSE,ICNT,IS,SAMP,RINCR,RARM,REND,
2      .ANGLE,STP)
3
4      C-----
5      C  THIS MOD CAPTURES THE DATA FOR THE POLARIZERS IN POSITION
6      C  VERTICAL/45 deg. TWO ARRAYS ARE USED TO SAVE THE DATA.
7      C  BRAY - IS A SEQUENTIAL ARRAY STORING ALL THE DATA RECEIVED FROM
8      C  THE A/D CONVERTER THRU THE ENTIRE ROTATION OF THE SAMPLE.
9      C
10     C  ERAY - IS A REPRESENTATION OF THE ENTIRE MATRIX. NINE ELEMENTS
11     C  WILL BE PASSED IN FROM THE A/D CONVERTER OF WHICH ONLY 3 WILL
12     C  BE NEW. THEY ARE : (9,10,12) FROM THE 1,2,4,9,10,12,13,14,16.
13     C  GAPS WILL BE LEFT IN THE ARRAY TO BE FILLED IN BY THE OTHER
14     C  TWO POSSIBLE POLARIZER SETTINGS.
15     C-----
16
17
18     CHARACTER SAMP*20,MSG*50,MESC*255,DIST*10,HEX*4
19     CHARACTER CHAN*2,PORT*10
20
21     REAL RDAT,ARAY,BRAY,CRAY,DRAW,ERAY,RINCR,REND,ANGLE,RARM
22
23     INTEGER STP,ICNG,REPORT,TYPE,GETDAT
24
25     DIMENSION GETDAT(16),RDAT(16),J1(3),HEX(16)
26
27     COMMON /MATRIX/ARAY(3600),BRAY(3600),CRAY(3600),DRAW(3600),
28     . ERAY(5800)
29
30     DATA J1 /9,10,12/      ! NEW MATRIX ARRAY ELEMENTS
31
32     ISWEEP - 2              ! FLAG THAT THIS IS 2ND SWEEP
33                             ! OF THE POLARIZERS
34     C-----
35     C  THIS MOVES THE POLARIZERS TO THE PROPER POSITION:
36     C  RECEIVER POLARIZER( AXIS 3 ) MOVES -225000 STEPS OR -45 DEGREES
37     C-----
38
39     WRITE(3, '(A4)' )'1X0 '      ! RESET CUMM POSITION COUNTER
40     CALL TWAIT(1)
41
42     WRITE(3, '(A38)' )'3P5 35T1 3V10 3A10 3D-225000 3G 3CR C '
43
44     READ(3, '(A50)' )MSG(1:50)
45
46     CALL TWAIT(1)
47
48     WRITE(3, '(A5)' )'35T0 '
49     CALL TWAIT(1)
50     WRITE(3, '(A13)' )'1P5 15T1 1H '
51

```

# Appendix IV

```

52      INUSE1 = 0
53      C _____
54      C   HERE BEGINS THE LOOP WHERE SAMPLE DATA IS TAKEN AND STORED
55      C _____
56
57      DO 200 M = 1, ISHOTS      ! LOOP THRU THE SAMPLE ROTATION
58
59      C _____
60      C   HERE IS WHERE THE A/D CONVERTER IS ASKED FOR THE DATA
61      C _____
62
63
64      C   CALL TESTDAT(ISWEEP, ANGLE, RDAT)
65
66      IC1 = 11
67      TYPE = 3
68      CALL DATEL(TYPE, REPORT, PORT, IC1, CHAN, RDAT, HEX)
69
70      C _____
71      C   HERE EACH ARRAY IS SELECTED, WRITTEN TO AND INCREMENTED
72      C _____
73
74      DO 100 I = 1, 10, 1, -1
75          ERAY(IS) = RDAT(I)      ! STORE THE 10 ARRAY ELEMENTS
76          IS = IS + 1      ! ARRAY SEQUENTIAL COUNTER
77      100  CONTINUE
78
79
80      C _____
81      C   HERE ONLY THREE ELEMENTS OF THE ERAY MATRIX ARE NEW. THEY ARE
82      C   ELEMENTS 9, 10, 12. THEY ARE STORED IN THE RDAT ARRAY IN POSITIONS
83      C   4, 5, 6 RESPECTIVELY. I USE I1 TO SET THE CORRECT SEQUENCE.
84      C _____
85
86      I1 = 3      ! START PT IN RDAT ARRAY
87      DO 150 I = 1, 3      ! LOOP THRU THE 3 NEW ELEMENTS
88          K = J1(I) + ICNT      ! SELECT CORRECT ARRAY ELEMENT
89          ERAY(K) = RDAT(I + I1) ! PLACE MATRIX DATA IN ERAY
90      150  CONTINUE
91
92      C _____
93      C   THIS OUTPUTS THE DATA EITHER BY GRAPHICS OR A/D VOLTAGES OR NONE
94      C   AT ALL DEPENDING ON THE FLAG STP.
95      C _____
96
97      IF (STP.EQ.1) THEN      ! REAL TIME TEX GRAPHICS
98
99          CALL DRAW_ELE(ISWEEP, ANGLE, RARM, RINCR, REND, RDAT, INUSE1)
100         GOTO 160
101     ENDR
102
103     IF (STP.EQ.3) GOTO 160      ! NO OUTPUT
104
105     CALL VIEW(SAMP, ANGLE, IS, STP, IDIR)

```

# Appendix IV

```

106
107 160 IF(M.EQ.ISHOTS)GOTO 200
108
109     ANGLE = ANGLE - RINCR    ! NEW SAMPLE ANGLE
110
111     ICNT = ICNT - 16        ! DECREMENT THE ARRAY BY 16
112
113
114 C-----
115
116     WRITE(3, '(A9)') '1G 1X1 C '
117     READ(3, '(A50)', ERR = 200) MESC(1:50)
118     CALL TWAIT(1)
119     READ(MESC(14:22), '(B1,19)', ERR = 200) MOTION
120     ICNG = MOTION - MBUF
121     MBUF = MOTION
122 170  FORMAT(5X,14,10X, ' ACCUMULATED MOTION: ',19,
123     ' RELATIVE MOTION: ',19//)
124 D     WRITE(*,170) M, MOTION, ICNG
125     WRITE(3, '(A4)') '1PS '
126     CALL TWAIT(1)
127 200  CONTINUE                ! LOOP THRU ROTATIONS
128
129 C-----
130 C   AT THIS POINT THE DATA COLLECTION FOR POLARIZERS POSITIONED AT
131 C   VERTICAL/45 deg IS COMPLETE. WE NOW RETURN TO THE CALLER
132 C   WHERE THE NEXT POLARIZER SETTING WILL BE MADE AND THE SAMPLE WILL
133 C   BE RETURNED TO THE START POSITION.
134 C-----
135
136     WRITE(3, '(A5)') '1ST0 '    ! DEENERGIZE SAMPLE MOTOR
137
138 400  RETURN
139     END

```

# Appendix IV

## AIV.30 Analog APSD Software Modules: VIEW Source Code.

```

1      SUBROUTINE VIEW(SAMP,ANGLE,IS,STP,DIR)
2
3      C -----
4      C   THIS ROUTINE BRINGS ALL THE ARRAY DATA TO THE SCREEN FOR EACH
5      C   ANGLE THAT DATA IS TAKEN FOR. THE REASON IS TO BE ABLE TO COMPARE
6      C   THE MATRIX DATA AS THRU THE FOUR POLARIZER CHANGES.
7      C -----
8
9      CHARACTER SAMP*20
10     REAL A,B,C,D,E,F,POS
11     INTEGER STP
12     COMMON /MATRIX/A(1600),B(1600),C(1600),D(1600),E(3000)
13
14     10  FORMAT (13X,'ANGLE: ',F7.2,25X,'SAMPLE: ',A20,/)
15     20  FORMAT(14X,'V/V',7X,'V',7X,'V/V',6X,'V',7X,'V/V',6X,
16         'V',6X,'V/V')
17     30  FORMAT(80X,' ')
18     40  FORMAT(4X,I4,4X,F7.5,3(5X,' ',5X,F7.5))
19     60  FORMAT(10X,'RETURN - CONT, < 99 > - DONT PAUSE,
20         '< 100 > - NO VIEW')
21         F = 0.00000
22
23     C -----
24     C   THE DATA SENT TO THE SCREEN IS DEPENDENT ON THE DIRECTION OF THE
25     C   SCAN OF THE LASER. IF THE SAMPLE IS GOING FROM LEFT TO RIGHT THEN
26     C   I SUBTRACT 9 FROM THE DATA ARRAY COUNTER. OTHERWISE ITS OK THE
27     C   WAY IT IS.
28     C -----
29
30     IF(DIR.EQ.1)IS = IS - 9
31
32     C -----
33     C   HERE THE ENTIRE SCREEN IS WRITTEN TOO WITH FOUR FIELDS SERVING
34     C   AS THE FOUR ARRAYS. IS - THE START OF THE MATRIX FOR THE
35     C   ANGLE SHOWING WHICH INCLUDES 16 ARRAY FIELDS.
36     C -----
37
38     50  WRITE(*,10)POS,SAMP
39         WRITE(*,20)
40         WRITE(*,30)
41
42         WRITE(*,40)1,A(1 + IS),B(1 + IS),C(1 + IS),D(1 + IS)
43         WRITE(*,40)2,A(2 + IS),B(2 + IS),F,F
44         WRITE(*,40)3,F,F,C(2 + IS),D(2 + IS)
45         WRITE(*,40)4,A(3 + IS),B(3 + IS),C(3 + IS),D(3 + IS)
46         WRITE(*,40)5,A(4 + IS),F,C(4 + IS),F
47         WRITE(*,40)6,A(5 + IS),F,F,F
48         WRITE(*,40)7,F,F,C(5 + IS),F
49         WRITE(*,40)8,A(6 + IS),F,C(6 + IS),F
50         WRITE(*,40)9,F,B(4 + IS),F,D(4 + IS)
51         WRITE(*,40)10,F,B(5 + IS),F,F

```

## Appendix IV

```

32      WRITE(*,40)11,F,F,F,D(5 + IS)
33      WRITE(*,40)12,F,B(6 + IS),F,D(6 + IS)
34      WRITE(*,40)13,A(7 + IS),B(7 + IS),C(7 + IS),D(7 + IS)
35      WRITE(*,40)14,A(8 + IS),B(8 + IS),F,F
36      WRITE(*,40)15,F,F,C(8 + IS),D(8 + IS)
37      WRITE(*,40)16,A(9 + IS),B(9 + IS),C(9 + IS),D(9 + IS)
38
39      C-----
60      C   THAT COMPLETES THE MATRIX VIEW. HERE IM ADDING A READ STMT SO
61      C   THE USER CAN OPT TO SEE HOW THE DATA IS COMMING IN. IF < 99 >
62      C   IS TYPED THIS READ WILL BE SKIPPED.
63      C   STP IS THE STOP FLAG. AS LONG AS ITS 2 THEN THE DATA WILL BE
64      C   WRITTEN EVERY TIME. IF < 99 > IS ISSUED THEN THE DATA WILL BE
65      C   WRITTEN BUT THE STP WONT STOP. IF < 100 > IS ISSUED THEN STP
66      C   WILL NOTIFY THE CALLER NOT TO CALL THIS SUBROUTINE. I'M LEAVING
67      C   AN OPTION FOR STP TO BE RESET TO ZERO BY THE CALLER(1st LINE BELOW)
68      C-----
69
70      IF(STP.EQ.2)INUSE = 0
71      IF(INUSE.EQ.0)THEN
72          WRITE(*,60)
73          READ*,'(A5)'STP
74          IF(STP.EQ.99)INUSE = 1
75      ENDIF
76
77
78      IF(MDIR.EQ.1)IS = IS + 9
79      RETURN
80      END

```

# Appendix IV

## AIV.31 Analog APSD Software Modules: VV Source Code.

```

1      SUBROUTINE VV(ISHOTS,INUSE,ICNT,IS,SAMP,RINCR,RARM,REND,
2      .ANGLE,STP,PORT)
3
4      C-----
5      C   THIS MOD CAPTURES THE DATA FOR THE POLARIZERS IN POSITION
6      C   VERTICAL/VERTICAL. TWO ARRAYS ARE USED TO SAVE THE DATA.
7      C   ARAY - IS A SEQUENTIAL ARRAY STORING ALL THE DATA RECEIVED FROM
8      C   THE A/D CONVERTER THRU THE ENTIRE ROTATION OF THE SAMPLE.
9      C
10     C   ERAY - IS A REPRESENTATION OF THE ENTIRE MATRIX. ONLY NINE
11     C   ELEMENTS WILL BE FILLED IN THIS ARRAY WITH THE OPTICS IN THIS
12     C   POSITION FOR EACH LASER SHOT. THEY ARE : 1,2,4,5,6,8,13,14,16.
13     C   GAPS WILL BE LEFT IN THE ARRAY TO BE FILLED IN BY THE OTHER
14     C   THREE POSSIBLE POLARIZER SETTINGS.
15     C   FEWER ELEMENTS WILL BE COLLECTED ON SUCCEEDING SWEEPS OF THE
16     C   GONIOMETER ARM.
17     C-----
18
19
20     CHARACTER STR*9,GETDAT*4,CR,SAMP*20,A*80,DIST*10
21     CHARACTER MSG*255,MSG*80,PORT*10,HEX*4
22     CHARACTER CHAN*2,CHANNEL*2
23
24
25     REAL RDAT,ARAY,BRAY,CRAY,DRAW,ERAY,RARM,ANGLE,RINCR,INC,REND
26
27     INTEGER POS,STP,ISIZE,IOUT,REPORT,TYPE
28     INTEGER*4 LENGTH,TIMEOUT,ICNG
29
30     DIMENSION GETDAT(16),J(9),RDAT(16),HEX(16)
31
32     COMMON /MATRIX/ARAY(3600),BRAY(3600),CRAY(3600),DRAW(3600),
33     .ERAY(3600)
34
35     DATA J /1,2,4,5,6,8,13,14,16/ ! MATRIX ARRAY ELEMENTS
36
37     ISWEEP - 1          ! FLAG THAT THIS IS 1ST SWEEP
38     ANGLE - RARM        ! ANGLE - START POSITION OF SAMP
39
40     C-----
41     C   INUSE IS PASSED IN AS ZERO EVERY TIME A NEW LASER IS ENABLED
42     C   SO THAT ALL THE COUNTERS OF THE ARRAYS ARE CORRECT
43     C-----
44
45     IF(INUSE.EQ.0)THEN      ! INITIALIZE VARIABLES
46         INUSE - 1          ! FLAG NOT TO COME BACK
47         ICNT - 0           ! 16 ELEMENT ARRAY COUNTER
48         IS - 0            ! SEQUENTIAL ARAY COUNTER
49     ENDIF
50
51     C-----

```

# Appendix IV

```

52 C THIS INITIALIZES THE SAMPLE STAGE MOVEMENT OF VELOCITY,
53 C ACCELERATION, AND DISTANCE.
54 C-----
55
56 INC = INCR * 2000. ! THIS IS THE ANGLE INCREMENT
57 INCR = IPX(INC)
58 NUMBER = 10
59
60 DO 30 I = 1,10
61 IP(INCR,LT,NUMBER)GOTO 60
62 NUMBER = NUMBER * 10
63 30 CONTINUE
64
65 C-----
66 C IM USING THIS TO CLEAR THE READ BUFFER
67 C-----
68
69 C PORT = 'TXA2'
70 60 LENGTH = 255
71
72 TIMEOUT = 1
73
74 CALL READ_QIO(PORT,MESC,LENGTH,TIMEOUT,ISIZE,IOUT,CHANNEL,IUS)
75
76 C-----
77
78 WRITEQ, '(A4)'1X0 ' ! RESET CUMM POSITION CNT
79
80 CALL MIX(INCR,DIST)
81
82 IF(INCR.LT.1000)THEN
83 WRITEQ, '(A27)'1P5 1ST1 1V10 1A10 1D-//DIST(1:1)/r '
84 ELSE
85 WRITEQ, '(A28)'1P5 1ST1 1V10 1A10 1D-//DIST(1:1)/r '
86 ENDP
87
88 C WRITEQ, '(A23)'1ST1 1V10 1A10 1D-//DIST(1:1)/r '
89
90 CALL TWAIT(30)
91
92 INUSE1 = 2 ! INITIALIZE TEK DRAW FLAG
93
94 C-----
95 C HERE BEGINS THE LOOP WHERE SAMPLE DATA IS TAKEN AND STORED
96 C-----
97
98 DO 200 M = 1,SHOTS ! LOOP THRU THE SAMPLE ROTATION
99
100 C-----
101 C HERE IS WHERE THE A/D CONVERTER IS ASKED FOR THE DATA
102 C-----
103
104
105 C CALL TESTDAT(SWEEP,ANGLE,RDAT)

```

# Appendix IV

```

106
107      IC1 = 11
108      TYPE = 3          ! STROBE THE A/D BOARD
109      CALL DATEL(TYPE,REPORT,PORT,IC1,CHAN,RDAT,HEX)
110                      ! GET HEX DATA FROM A/D BOARD
111
112      C-----
113      C   HERE EACH ARRAY IS SELECTED, WRITTEN TO AND INCREMENTED
114      C-----
115
116      DO 100 I=1,10
117          IS = IS + 1      ! ARRAY SEQUENTIAL COUNTER
118          ARAY(IS) = RDAT(I) ! STORE THE 10 ARRAY ELEMENTS
119          K = J(I) + ICNT ! SELECT CORRECT ARRAY ELEMENT
120          IF(I.EQ.10)GOTO 100 ! DO NOT STORE THE LAST DC ELE.
121          ERAY(K) = RDAT(I) ! PLACE MATRIX DATA IN ERAY
122      100 CONTINUE
123
124      C-----
125      C   THIS PART DISPLAYS OUTPUT IN EITHER A/D VOLTAGES OR GRAPHICS
126      C   THE CALLER WILL DEFINE STP AS 1 FOR GRAPHICS, 2 FOR A/D VOLTAGES
127      C   OR 3 NOT YET DEFINED.
128      C-----
129
130      IF(STP.EQ.1)THEN      ! USER WANTS REAL GRAPHICS
131
132          CALL DRAW_ELE(ISWEEP,ANGLE,RARM,RINCR,REND,RDAT,INUSE1)
133          GOTO 150
134
135      ENDF
136
137      IF(STP.EQ.3)GOTO 150 ! SKIP VIEW FLAG
138
139          CALL VIEW(SAMP,ANGLE,IS,STP,IDIR)
140
141      C-----
142
143
144      150  IF(M.EQ.ISHOTS)GOTO 200 ! DO NOT MOVE SAMPLE ON LAST SHOT
145
146          ANGLE = ANGLE + RINCR      ! INCREMENT ANGLE OF SAMPLE
147
148          ICNT = ICNT + 16          ! INCREMENT THE ARRAY BY 16
149
150      C-----
151
152          WRITE(3, '(A9)')1G 1X1 C '
153
154          READ(3, '(A30)',ERR=200)MSG(1:30)
155
156      D   WRITE(*,*)M, ' MSG(1:30) = ',MSG(1:30)
157
158          CALL TWAFT(1)
159
159

```

# Appendix IV

```

160      READ(MESG(14:22),'(BN,P)',ERR = 200)MOTION
161
162      ICNG = MOTION - MBUF
163      MBUF = MOTION
164
165 160  FORMAT(5X,I4,10X,' ACCUMULATED MOTION: ',P,
166      ' RELATIVE MOTION: ',P//)
167
168  D      WRITE(' ',160)M,MOTION,ICNG
169      WRITE(3,'(A4)')'1P5 '
170      CALL TWAIT(1)
171
172  C_____
173
174 200  CONTINUE          ! LOOP THRU ROTATIONS
175
176  C_____
177  C      AT THIS POINT THE DATA COLLECTION FOR POLARIZERS POSITIONED AT
178  C      VERTICAL/VERTICAL IS COMPLETE. WE NOW RETURN TO THE CALLER
179  C      WHERE THE NEXT POLARIZER SETTING WILL BE MADE AND THE SAMPLE WILL
180  C      BE ROTATED IN REVERSE.
181  C_____
182
183      WRITE(' ',(A5))'1ST0 '      ! DEENERGIZE SAMPLE STAGE
184
185 400  RETURN
186      END

```

## Appendix IV

### AIV.32 Analog APSD Software Modules: YHI Source Code.

```

1      SUBROUTINE YHI(A,X,Y,BUTT)
2
3      C _____
4      C   THIS MOD CHANGES PACKED TEK 4957 CODES TO THEIR X,Y EQUIV.
5      C   WHEN A GIN DEVICE IS ACTIVATED IT SENDS TO THE HOST A 6 OR 12
6      C   CHARACTER LENGTH STRING. THE FIRST CHARACTER IS A NUMBER OR
7      C   CHARACTER REPRESENTING THE KEY OR BUTTON PRESSED. AS A MOUSE HAS
8      C   THREE BUTTONS THE FIRST CHARACTER WOULD BE A 1, 2 OR 3.
9      C   THE NEXT 5 CHARACTERS ARE THE X,Y SCREEN COORDINATES.
10     C   THIS ROUTINE PARSES THE FIRST SIX CHARACTERS THE FIRST IS
11     C   RETURNED IN BUTT AND THE X,Y ARE RETURNED AS INTEGERS.
12     C       IT IS INTERESTING TO NOTE THAT THE FIVE CHARACTER STRING
13     C   THAT REPRESENTS X,Y CANNOT BE USED DIRECTLY IN OTHER COMMANDS.
14     C   THE LOCATIONS WOULD NOT BE THE SAME. SO THE RETURNED X,Y THEN
15     C   HAS TO BE PASSED BACK THRU HIY. THUS THE NAME OF THIS ROUTINE
16     C   IS THE MIRROR OF HIY...YHI.
17     C   THE MULTIPLICATIONS ARE FOR BIT SHIFTING
18     C
19     C _____
20
21     CHARACTER *6 A,BUTT*1
22     INTEGER X,Y
23
24
25     BUTT=A(1:1) ! THIS IS USUALLY THE THE BUTTON NUMBER
26     X=(ICHAR(A(3:5))-32) * 128 + (ICHAR(A(6:6))-32)*4
27     $ + MOD(ICHAR(A(3:3)),4)
28     Y=(ICHAR(A(2:2))-32) * 128 + (ICHAR(A(4:4))-32)*4
29     $ + MOD(ICHAR(A(3:3))/4,4)
30     RETURN
31     END

```

Blank

## APPENDIX V: RETRO/DISPLAY USER GUIDES AND SOURCE CODES.

RETRO/DISPLAY are numeric and graphic implementations of Full-Wave electromagnetic scattering theory. The theory was developed by Professor Ezekiel Bahar at the University of Nebraska-Lincoln. The reader is advised to read publications 30-34 listed in the Literature Cited section for details about this theory. Equations notated [A; B] imply that A is the Literature Cited reference number, and B the page number from that reference. The Full Wave code was written for a CRAY supercomputer operating under UNIX UNICOS.

### AV.1 INPUT FILES.

Two input files are needed by RETRO. The first is "params 1," which is used in every run and contains filenames, surface information, and integration routine controls. The second "dielectric's input" file contains information to determine the material's relative dielectric constant,  $\epsilon_r$ , as a function of wavelength,  $\lambda_0$ .

#### AV.1.1 "params."

The following information, in order, is stored in file "params": filename for commentary output, filename for data output, a 13 character description of the surface material, desired mean squared heights, mean squared slopes, radiation wavelengths, incident angles, code for getting  $\epsilon_r$  data, and the error requirements for IMSL integration routines, QDAG and TWODQ. An example of a correctly edited file is shown at the end of this guide. The line by line format is as follows:

- A - filename for commentary output (up to 13 characters)
- B - filename for data output (up to 13 characters)
- C - description of surface material (up to 13 characters)
- D - no. of  $\langle h^2 \rangle$ 's,  $\min(\langle h^2 \rangle)$ ,  $\text{step}(\langle h^2 \rangle)$  in  $\mu m^2$  (integer,real,real)
- E - lines E through F exist only if the no. of  $\langle h^2 \rangle$ 's  $< 0$
- F - here the min and step values are ignored in favor of reading in a list of  $\langle h^2 \rangle$  values separated by spaces
- G - no. of  $\sigma_z^2$ 's,  $\min(\sigma_z^2)$ ,  $\text{step}(\sigma_z^2)$  (integer,real,real)
- H - same as for  $\langle h^2 \rangle$ 's
- I -
- J - no. of  $\lambda_0$ 's,  $\min(\lambda_0)$ ,  $\text{step}(\lambda_0)$  in  $\mu m$  (integer,real,real)
- K - same as for  $\langle h^2 \rangle$ 's
- L -
- M - no. of  $\theta_0$ 's,  $\min(\theta_0)$ ,  $\text{step}(\theta_0)$  in degrees (integer,real,real)
- N - same as for  $\langle h^2 \rangle$ 's
- O -
- P - A code for determining  $\epsilon_r$  values (integer = 0 or 1)
- Q - if the code on line P is 1 then the relative dielectric is put here (complex)
  - otherwise the filename containing the  $\epsilon_r$ 's is here (up to 13 characters)
- R - absolute and relative error criteria for QDAG (real,real)
- S - absolute and relative error criteria for TWODQ (real,real)

The data on lines A,B,C,D,G,J,M,P,Q,R, and S must appear in "params." The data listed on lines E,F,H,I,K,L,N, and O may or may not be needed, depending on the sign of the first number on lines D,G,J, and M.

### AV.1.2 Dielectrics Input File.

This file passes RETRO the information to calculate  $\epsilon_r = \epsilon_r(\lambda_0)$ . The filename is passed on line Q in the above format. The usual form of dielectric filenames is "material.nk," where "material" is a singular name for the substance making up the surface. The ".nk" implies that the real and imaginary part of the relative index of refraction,  $n_r = n + ik$ , where  $n = n(\lambda_0)$  and  $k = k(\lambda_0)$ , is passed instead of  $\epsilon_r = n_r^2$ . The data is stored this way because  $n$  and  $k$  values are easier to find in tables. The format for "material.nk" is as follows:

A - the number of wavelengths in list, m (integer)

B -  $\lambda_0(1), n(1), k(1)$  (real,real,real)

.

Z -  $\lambda_0(m), n(m), k(m)$  (real,real,real)

where  $\lambda_0(l) < \lambda_0(l+1)$ . Also, for the interpolation routine to work,  $m > 6$ . A sample file is shown at the end of this guide.

### AV.2 OUTPUT FILES.

Two output files are created by RETRO, a commentary output and the data output. These filenames are defined on lines A and B of "params." The commentary file is primarily a debugging tool should something go wrong. The computed matrices and a header block are put in the data file.

The commentary output file is opened as unit 8 in RETRO. Unit 8 is then sent various values for tracing and debugging the program. The file becomes important if unknown run-time errors occur. Most of the write statements to unit 8 have been commented out, but they can be reinstated by editing RETRO and changing some "c"'s to blanks. During normal operation, the only significant information sent to this file is the list of  $\epsilon_r$ 's for each  $\lambda_0$ 's after interpolation from the dielectric file. DISPLAY does not access this file for plotting.

#### AV.2.1 Data Output File.

This file contains two distinct sections: the heading block and the data block. The heading block is the key to finding the elements of Mueller matrix,  $F(<h^2>, \sigma_z^2, \lambda_0, \theta_0)$ . The format of the heading block is as follows :

A - description of material ("params" line C)

B - no. of  $<h^2>$ 's,  $\sigma_z^2$ 's,  $\lambda_0$ 's, and  $\theta_0$ 's ("params" lines D,G,J, and M)

C -  $<h^2>(1) \dots <h^2>(\text{no. of } <h^2>\text{'s})$  ("params" line D or lines E and F)

D -  $\sigma_z^2(1) \dots \sigma_z^2(\text{no. of } \sigma_z^2\text{'s})$  ("params" line G or lines H and I)

E -  $\lambda_0(1) \dots \lambda_0(\text{no. of } \lambda_0\text{'s})$  ("params" line J or lines K and L)

F -  $\theta_0(1) \dots \theta_0(\text{no. of } \theta_0\text{'s})$  ("params" line M or lines N and O)

The data block consists of two data lines per matrix. The first line contains three Q values corresponding to the auto-correlation functions Gaussian, N=8, and N=6. The second line contains six generally non-zero matrix elements divided by Q. In order, these correspond to  $f_{11}$ ,  $f_{12}$ ,  $f_{22}$ ,  $f_{33}$ ,  $f_{34}$ , and  $f_{44}$ . RETRO does not write  $f_{21} = f_{12}$  or

$f_{43} = -f_{34}$ . The other 8 matrix elements are zero.

RETRO calculates and writes the matrix data in a nested loop where  $\langle h^2 \rangle$  varies more slowly than  $\sigma_z^2$ , which varies more slowly than  $\lambda_0$ , which varies more slowly than  $\theta_0$ . The looping of each variable goes from the first value to the last value listed in the header (lines C to F). Given this information, DISPLAY can access any of the F's in the file.

### AV.3 RUNNING INSTRUCTIONS.

RETRO is ready for execution after the input file "params" has been properly edited and the dielectric file for the surface material has been loaded. There are two ways of running the program, and the preferred method is based on how many matrices must be calculated. The number of Mueller matrices calculated is the number of  $\langle h^2 \rangle$ 's times the number of  $\sigma_z^2$ 's times the number of  $\lambda_0$ 's times the number of  $\theta_0$ 's. A short run is considered less than 100 matrix elements.

#### AV.3.1 Short Interactive Runs.

Runs that calculate less than 100 Mueller matrices can be done interactively by typing "retro.run" after the prompt. This command will compile, link, and execute the program using the data provided in file "params." This executable code is stored in file "retro.xqt." If you desire to run the program again at the same login after changing "params," you need only type "retro.xqt" after the prompt. This should be done instead of "retro.run" to prevent unnecessary compiling and linking. Finally, before exiting, you should remove "retro.xqt." The object file "retro.o" was removed by "retro.run" immediately following the linking procedure.

#### AV.3.2 Long Batch Runs.

For longer runs, a job needs to be submitted to a batch queue. For runs of less than 2000 matrices, type "qsub -eo -q small2 examp." If the number of matrices is greater than 2000, replace "small2" with "large2" in the previous string. Immediately after the submit command, the computer returns the job's assignment number, N. Write this down, and if you discover an error in "params" while the job is still running you can stop it by giving the command "qdel -k N," where N is the assignment number.

Files "examp1" and "examp2" are the run streams for the batched job. Only one run stream named "examp1" is allowed in the batch queue at a time. However, it is possible to have more than one batch job running at a time if copies of examp1 are submitted in exactly the same way. The file "params" is read soon after the submit command, so it can be edited and the next job submitted while the first is still active. The same comments about removing "retro.xqt" apply here as in the previous section.

When a batch job is finished, an additional output file is created under the name examp1.oN, where N is defined as before. This is the job accounting report for the run, that includes the job's user time, system time, and billing units. For a complete description see the Cray manual's "jar" (job accounting report generation) command. A good idea is to append this accounting report to the commentary file created by the run. This is the file named on line B of "params."

## Appendix V

### AV.4 RELATIONSHIP OF PROGRAM VARIABLES TO THEORY.

The following is a break down of the various theoretical parameters followed by the relationship of program variables to these values. See reference 26 for further explanation.

#### AV.4.1 Computing Q:

$$Q = \frac{k_0^2}{\pi} Q'$$

$$Q' = 2\pi \int_0^{\infty} (\chi_2(r_d) - |\chi|^2) J_0(v_{xz} r_d) r_d dr_d$$

$$\chi_2 = \exp(v_y^2 \langle h^2 \rangle (1 - r_{hh}))$$

$$v_y = 2k_0 \cos(\theta_0)$$

$$r_{hh} = r_{hh}(r_d)$$

$$|\chi|^2 = \exp(-v_y^2 \langle h^2 \rangle)$$

$$v_{xz} = |2k_0 \sin(\theta_0)|$$

$Q = Q$   
 $K0SQ = k_0^2$   
 $PI = \pi$   
 $RD = r_d$   
 $QPRIME = Q'$   
 $CHI2 = \chi_2$   
 $VYSQH = v_y^2 \langle h^2 \rangle$   
 $VY = v_y$   
 $HMSQ = \langle h^2 \rangle$   
 $K0 = k_0$   
 $THETA = \theta_0$   
 $CSTHT = \cos(\theta_0)$   
 $SNHT = \sin(\theta_0)$   
 $RRSURF(RD) = r_{hh}(r_d)$   
 $CHISQ = |\chi|^2$   
 $JSUB0 = J_0(v_{xz} r_d)$   
 $VXZ = v_{xz}$

#### AV.4.2 Computing $r_{hh}$ .

# Appendix V

For the Gaussian auto-correlation function:

$$r_{hh}(r_d) = \exp\left(\frac{-r_d^2}{l_c^2}\right)$$

For the N=8 auto-correlation function:

$$r_{hh}(r_d) = \left[1 - \frac{3\zeta^2}{8} + \frac{\zeta^4}{32} + \frac{\zeta^6}{3072}\right] \zeta K_1(\zeta) + \left[\frac{1}{2} - \frac{\zeta^2}{4} - \frac{\zeta^4}{96}\right] \zeta^2 K_0(\zeta) \quad ([33;45])$$

$$\zeta = r_d \kappa_8, \quad \kappa_8^2 = \frac{.4 \sigma_S^2}{\langle h^2 \rangle}$$

For the N=6 auto-correlation function:

$$r_{hh}(r_d) = \left[1 - \frac{3\zeta^2}{4} - \frac{\zeta^4}{96}\right] \zeta K_1(\zeta) + \left[\frac{1}{2} + \frac{3\zeta^2}{16}\right] \zeta^2 K_0(\zeta)$$

$$\zeta = r_d \kappa_6, \quad \kappa_6^2 = \frac{\sigma_S^2}{\langle h^2 \rangle}$$

CLENSQ =  $l_c^2$   
R =  $\zeta$   
RSQ =  $\zeta^2$   
R4TH =  $\zeta^4$   
R6TH =  $\zeta^6$   
KAPPA8 =  $\kappa_8$   
HMSQ =  $\langle h^2 \rangle$   
SIGS =  $\sigma_S^2$   
KAPPA6 =  $\kappa_6$

AV.4.3 Computing  $\sigma_{ij}^{kl}/Q$ :

$$\frac{\sigma_{ij}^{kl}}{Q} = 2 \int_{h_x=0}^{\infty} \int_{h_z=-\infty}^{\infty} \frac{D^{ij} D^{kl}}{(\bar{n} \cdot \bar{a}_y)^2} P_2 P_S dh_x dh_z \quad ([32;731-732])$$

# Appendix V

$$P_2(h_x) = \frac{1}{1+C_0} U(\cotan(\theta_0))$$

$$C_0 = \frac{1}{2} \frac{\sqrt{\sigma_z^2}}{\sqrt{\pi}} \tan(\theta_0) \exp\left(-\frac{\cotan^2(\theta_0)}{\sigma_z^2}\right) - \frac{1}{2} \operatorname{erfc}\left(\frac{\cotan(\theta_0)}{\sqrt{\sigma_z^2}}\right)$$

$$P_S(h_x, h_z) = \frac{1}{\pi \sigma_z^2} \exp\left(-\frac{h_x^2 + h_z^2}{\sigma_z^2}\right)$$

$$\text{IDD} = \sigma_{ij}^{kl}$$

$$\text{DPQMAG} = \frac{D^{ij} D^{kl}}{(\vec{n} \cdot \vec{a}_y)^2}$$

$$P2 = P_2$$

$$\text{HX} = h_x$$

$$\text{HZ} = h_z$$

$$\text{COTT} = \cotan(\theta_0)$$

$$\text{FACT} = \frac{1}{1+C_0}$$

$$\text{SIGSRT} = \sqrt{\sigma_z^2}$$

$$\text{PIC} = \frac{1}{2} \frac{\sqrt{\sigma_z^2}}{\sqrt{\pi}}$$

$$R = \frac{\cotan(\theta_0)}{\sqrt{\sigma_z^2}}, \quad (\text{in function ARGIDD only})$$

$$F1 = \exp\left(-\frac{\cotan^2(\theta_0)}{\sigma_z^2}\right)$$

$$F2 = \operatorname{erfc}\left(\frac{\cotan(\theta_0)}{\sqrt{\sigma_z^2}}\right)$$

$$C0 = C_0$$

$$\text{FHXHZ} = P_S$$

$$\text{HXZSQ} = h_x^2 + h_z^2$$

$$\text{AV.4.4 Computing } \frac{D^{ij} D^{kl}}{(\vec{n} \cdot \vec{a}_y)^2}:$$

$$\vec{n} \cdot \vec{a}_y = \cos(\gamma)$$

([31;446])

$\gamma$  = the angle between  $\vec{a}_y$  and the normal to the local scatter plane,  $\vec{n}$ .

# Appendix V

$$D^{PQ} = \begin{cases} D_{11}, & PQ = VV \\ D_{12}, & PQ = VH \\ D_{21}, & PQ = HV \\ D_{22}, & PQ = HH \end{cases}$$

$$D = \cos(\theta_0^n) T F T$$

$\theta_0^n$  = the angle between the incident direction and  $\vec{n}$ .

$$T = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} \quad ([31;448])$$

$\psi$  = the angle between the reference and local planes of incidence.

$$F = \begin{bmatrix} F^{VV} & 0 \\ 0 & F^{HH} \end{bmatrix}$$

$$CSG = \cos(\gamma)$$

$$DPQ(i,j) = D_{ij}$$

$$CS0N = \cos(\theta_0^n)$$

$$SN0N = \sin(\theta_0^n)$$

$$CSSI = \cos(\psi)$$

$$SNSII = \sin(\psi)$$

$$CFVV = \cos(\theta_0^n) F^{VV}$$

$$CFHH = \cos(\theta_0^n) F^{HH}$$

AV.4.5 Computing  $F^{VV}$  and  $F^{HH}$ :

$$F^{VV} = \frac{(\mu_r \cos^2(\theta_1^n) - \sin^2(\theta_0^n))(1 - \frac{1}{\epsilon_r}) + (1 - \mu_r)}{(\cos(\theta_0^n) + \eta_r \cos(\theta_1^n))^2} \quad ([31;446])$$

# Appendix V

$$F^{HH} = \frac{(\epsilon_r \cos^2(\theta_1^n) - \sin^2(\theta_0^n))(1 - \frac{1}{\mu_r}) + (1 - \epsilon_r)}{(\cos(\theta_0^n) + \frac{1}{\eta_r} \cos(\theta_1^n))^2}$$

$$\mu_r = \frac{\mu_1}{\mu_0}$$

$$\epsilon_r = \frac{\epsilon_1}{\epsilon_0}$$

$$\eta_r = \frac{\sqrt{\mu_r}}{\sqrt{\epsilon_r}}$$

$$n_r = \sqrt{\epsilon_r \mu_r}$$

$$\sin(\theta_1^n) = \frac{\sin(\theta_0^n)}{n_r}$$

$$\cos^2(\theta_1^n) = 1 - \sin^2(\theta_1^n)$$

$$ER = \epsilon_r$$

$$UR = \mu_r$$

$$RIR = n_r$$

$$ETAR = \eta_r$$

$$ERM = 1 - \epsilon_r$$

$$URM = 1 - \mu_r$$

$$ERMR = 1 - \frac{1}{\epsilon_r}$$

$$URMR = 1 - \frac{1}{\mu_r}$$

$$SN1N = \sin(\theta_1^n)$$

Appendix V

$$CS1N = \cos(\theta_1^n)$$

$$DEN2 = \cos(\theta_0^n) + \eta_r \cos(\theta_1^n)$$

$$DEN3 = \cos(\theta_0^n) + \frac{1}{\eta_r} \cos(\theta_1^n)$$

$$S0IF = \sin^2(\theta_0^n)$$

$$C1IF = \cos^2(\theta_1^n)$$

## Appendix V

### AV.5 RETRO SOURCE CODE.

```
1  C
2  c This work was done for the CRDEC on the Aberdeen Proving Grounds
3  c Edgewood Area.
4  c This work was done by Craig M. Herzinger under contract 88MQ450.
5  c
6  c This program uses the Full-wave Theory for computing the scattering
7  c of a plane wave from a randomly rough surface.
8  c
9  c This program is for SINGLE scatter from an ISOTROPIC rough surface.
10 c This program is for BACKSCATTER only.
11 c This program is for DIFFUSE scattering only.
12 c
13 c Single scatter implies the reflected radiation struck the rough
14 c surface only one time and that multiple scattering is unaccounted for.
15 c An isotropic surface is considered to invariant to rotation and translation
16 c in terms of the average scattering.
17 c Backscatter implies the receiver and detector are at the same point,
18 c with the same orientation, at a point far from the surface.
19 c Only diffuse scattering is calculated because the coherent specular term
20 c which occurs at normal incidence for backscatter is dropped.
21 c
22 c This program calculates the 8 generally non-zero Mueller Mtx elements,
23 c P11,12,21,22,33,34,43, and 44, for use with the standard Stokes Vector
24 c notation. Of these eight two pairs are degenerate, P12-P21 and P34-P43.
25 c The elements are calculated on a per solid angle basis so that the calculated
26 c Mueller Mtx is absolutely correct to within a scalar constant. The scalar
27 c constant is based upon the size of the solid angle intercepted by the
28 c detector for a particular experimental setup. For this work to be valid
29 c the detector must look at range of returned angles close enough to
30 c pure backscatter that the backscattered return is a good approximation
31 c of the entire reflected range. Also, the solid angle intercepted by the
32 c detector must be invariant to incident angle and wavelength.
33 c
34 c The program first calculates the scattering mbr, S, for use with the
35 c modified Stokes Vector notation, and this is then transformed into the
36 c desired form.
37 c
38 c The elements of S can be written as a product of two values, Q, and
39 c a 2-d slope averaging integral. This is allowed by the Full-wave
40 c Theory ONLY because slopes and heights are considered uncorrelated.
41 c Q is a function of incident angle, surface height auto-correlation
42 c function, and free space radiation wavelength. The slope integral
43 c is function of incident angle, mean squared slope, and wavelength
44 c through the surfaces relative dielectric constant.
45 c
46 c For the assumed isotropic surface, Q, normally a 2-d integral can
47 c be rewritten in polar coordinates and transformed into a 1-d integral.
48 c Q0 is computed by subroutine QCMP for 3 different spectral density/
49 c surface height auto-correlation functions. The inputs for the auto-corr.
50 c functions are mean squared height and mean squared slope. The 3
51 c functions are Gaussian, N-8, and N-6.
52 c
```

## Appendix V

```

53      c The slope averaging integrals account for all possible combinations of
54      c slopes in the x and z directions. Considering various polarizations and
55      c phase relationships, 16 unique integrals are possible to complete S.
56      c But 8 integrals -> 0 because the integrand is odd.
57      c 3 others converge to 1 value, and another -> 0 because the integrand
58      c is proportional to the imaginary part of a real number.
59      c This leaves 5 unique 2-d integrals performed by subroutine IDDCMP.
60      c The 5 values could be used compute S but they are recombined in one
61      c step to form the Mueller mtr elements, Pij.
62      c
63      c The following version 10.0 IMSL routines are required:
64      c   QDAG, TWODQ, B5J0, BSK0, BSK1, and ERPC.
65      c These are available from IMSL, customer relations, sixth floor,
66      c NBC Building 7300 Bellaire Boulevard, Houston, Texas 77036-5085, USA.
67      c Telephone (713)772-7927   Telex: 79-1923 IMSL INC HOU
68      c
69      c
70      c Input unit 7 is assigned to a file params that must contain the
71      c following input parameters:
72      c line01 - filename for program commentary from unit 8
73      c line02 - filename for output data from unit 9
74      c line03 - character*13 description of material being tested
75      c line04 - range or list of desired mean squared heights (in u^2)
76      c   - range or list of desired mean squared slopes (unitless)
77      c   - range or list of desired wavelengths (in u)
78      c   - range or list of desired incident angles (in deg.)
79      c     measured with respect to the normal of the reference plane
80      c   - prompt for dielectric of material
81      c   - the complex relative dielectric constant of the surface or
82      c     the name of the file containing list of constants
83      c   - integration requirements for QDAG
84      c   - integration requirements for TWODQ
85      c
86      c Unit 8 is sent various values for debugging the program and determining
87      c where something may have gone wrong. It sends values for monitoring the
88      c integration routines. Most writes to unit 8 have been commented out, but
89      c can be reinstated should the program have run time problems.
90      c
91      c Unit 9 is sent a heading block to access the Mueller Mtr data that follows.
92      c Elements of a Mueller Mtr for a given mean squared height, mean squared slope,
93      c radiation wavelength, and incident angle are printed on two lines.
94      c The first line contains the scalar Q values for the 3 different surface
95      c correlation functions. The second lines contains six of mtr elements
96      c divided by Q. The elements in order are P11,12,22,33,34 and 44.
97      c Any program making use of this output file must make the proper combinations
98      c and assignments to complete the total Mueller Mtr.
99      C
100     C
101     REAL PI,HMSQB,HMSQS,SIGSB,SIGSS,WLENB,WLENS,ANCB,ANGS,
102     1   HMSQ,CLEN,WLEN,CLENSQ,SIGS,K0,K0SQ,CSTHT,SNHTT,THETA,
103     2   HMSQL(15),SIGSL(15),WLENL(100),ANCL(100),
104     3   IDD(16),XS(16),P(16),Q(3),
105     4   AERR1,KERR1,AERR2,KERR2
106     REAL SETER

```

# Appendix V

```

107      REAL SVIDD(100,16)
108      C
109      COMPLEX ERCMP,ER
110      C
111      INTEGER NANG,NWLEN,NSICS,NHMSQ,II,I2,I3,I4,IL
112      I      DCODE,RCODE,IFLAG1
113      C
114      CHARACTER*15 OUTNM,COMNM
115      CHARACTER*13 MATNM
116      C
117      COMMON/ONE/HMSQ,CLEN,CLENSQ,WLEN,THETA,SNTHT,CSTHT,SIGS,K0,PI
118      C
119      C
120      PI=ABS(ATAN2(0.,-1.))
121      C
122      C This block opens file params
123      C It reads file names and opens files for commentary output and data output
124      C It also reads in the identifying material name and puts in output
125      OPEN(UNIT=7,FILE='params')
126      READ(7, '(A)')COMNM
127      OPEN(UNIT=8,FILE=COMNM)
128      WRITE(8, 'COMMENTARY FOR RETRO')
129      READ(7, '(a)')OUTNM
130      OPEN(UNIT=9,FILE=OUTNM)
131      READ(7, '(a)')MATNM
132      write(8, 'Commentary for retro')
133      write(8, 'Output file is ',OUTNM)
134      write(8, 'Material description is ',MATNM)
135      C
136      C
137      C THESE FOUR BLOCKS READ IN THE SURFACE AND BACKSCATTER PARAMETERS
138      C IF THE FIRST VAL IS NEGATIVE, A LIST OF VALUES IS TO FOLLOW
139      C IF THE FIRST VAL IS POSITIVE, A RANGE WITH THAT MANY VALUES IS SPECIFIED
140      C THIS ROUTINE THEN EITHER READS THE LIST INTO AN ARRAY OR
141      C FILLS THE ARRAY USING THE BEGINNING AND STEP VALUES GIVEN
142      READ(7, 'NHMSQ,HMSQB,HMSQS')
143      IF(NHMSQ.LT.0) THEN
144          NHMSQ=-1*NHMSQ
145          READ(7, '(HMSQL(IL),IL=1,NHMSQ)')
146      ELSE
147          DO 10, IL=1,NHMSQ
148              HMSQL(IL)=HMSQB+(IL-1)*HMSQS
149          ENDIF
150          WRITE(8, 'NHMSQ: ', 'MEAN SQUARE HEIGHT VALUES READ IN')
151      C
152      READ(7, 'NSICS,SIGSB,SIGSS')
153      IF(NSICS.LT.0) THEN
154          NSICS=-1*NSICS
155          READ(7, '(SIGSL(IL),IL=1,NSICS)')
156      ELSE
157          DO 20, IL=1,NSICS
158              SIGSL(IL)=SIGSB+(IL-1)*SIGSS
159          ENDIF
160          WRITE(8, 'NSICS: ', 'MEAN SQUARE SLOPE VALUES READ IN')

```

# Appendix V

```

161 C
162 READ(7,'NWLEN,WLENB,WLENS
163 IF(NWLEN.LT.0)THEN
164   NWLEN=-1*NWLEN
165   READ(7,'(WLENL(IL),IL-1,NWLEN)
166 ELSE
167   DO 30, IL-1,NWLEN
168 30 WLENL(IL)=WLENB+(IL-1)*WLENS
169 ENDIF
170 WRITE(8,'NWLEN,' ',WAVELENGTHS READ IN'
171 C
172 READ(7,'NANG,ANGB,ANGS
173 IF(NANG.LT.0)THEN
174   NANG=-1*NANG
175   READ(7,'(ANGL(IL),IL-1,NANG)
176 ELSE
177   DO 40, IL-1,NANG
178 40 ANGL(IL)=ANGB+(IL-1)*ANGS
179 ENDIF
180 WRITE(8,'NANG,' ',INCIDENT ANGLES READ IN'
181 C
182 C
183 C THIS SECTION PLACES A HEADING BLOCK INTO THE OUTPUT FILE
184 C THE SURFACE AND BACKSCATTER PARAMETERS ARE WRITTEN INTO THE OUTPUT
185 C FILE SO THAT LATER ANALYSIS ROUTINES WILL HAVE AN INDEX TO THE DATA
186   WRITE(9,'MATNM
187   WRITE(9,1000)NHMSQ,NSICS,NWLEN,NANG
188 1000 FORMAT(4F10.4)
189   WRITE(9,1010)(HMSQL(IL),IL-1,NHMSQ)
190   WRITE(9,1010)(SICSL(IL),IL-1,NSICS)
191   WRITE(9,1010)(WLENL(IL),IL-1,NWLEN)
192   WRITE(9,1010)(ANGL(IL),IL-1,NANG)
193 1010 FORMAT(3E12.4)
194 C
195 C
196 C THIS BLOCK READS IN THE CODE FOR THE RELATIVE DIELECTRIC CONSTANTS TO BE
197 C USED. THIS CODE AND THE DEGREES OF LAGRANGIAN INTERPOLATION TO BE USED
198 C IS PASSED TO ENTRY SETER OF FUNCTION ERCMP. THE NEXT LINE OF INPUT IS
199 C READ INSIDE ENTRY SETER.
200   READ(7,'DCODE
201   write(8,'Dielectric code read in ',dcode
202   SETUP=SETER(DCODE,5)
203 C
204 C
205 C THIS LAST INPUT DATA IS THE RELATIVE AND ABSOLUTE ERROR REQUIREMENTS TO BE
206 C BY THE INTEGRATION ROUTINES QDAC AND TWODQ RESPECTIVELY.
207   READ(7,'AERR1,RERR1
208   write(8,'Error criteria for QDAC read in ',aerr1,rerr1
209   READ(7,'AERR2,RERR2
210   write(8,'Error criteria for TWODQ read in ',aerr2,rerr2
211 C
212 C
213 C THESE FOUR NESTED LOOPS MAKE ALL COMBINATIONS OF SURFACES WITH
214 C WAVELENGTHS AND ANGLES OF INCIDENCE SPECIFIED.

```

## Appendix V

```

215   C THE DESIRED PARAMETERS ARE USED TO COMPUTE THE MUELLER
216   C MATRIX ELEMENTS BY FIRST CALCULATING A SCALING 1-D INTEGRAL, Q,
217   C AND THEN CALCULATING 16 2-D INTEGRALS, IDD1..IDD16, THAT ARE COMBINED IN THE
218   C CORRECT MANNER TO GIVE THE MUELLER MATRIX ELEMENTS FOR THE
219   C STANDARD STOKES VECTOR NOTATION.
220   C IN REALITY FOR THIS WORK ONLY 5 2-D INTEGRALS NEED BE CALCULATED BUT
221   C THE PROGRAM IS SET UP GENERALLY.
222       DO 30 I1=1,NHMSQ
223           HMSQ=HMSQ(I1)
224   C
225       DO 60 I2=1,NSIGS
226           SIGS=SIGS(I2)
227   C The correlation length can be calculated when hmaq and sigs are fixed
228       CLENSQ=4.*HMSQ/SIGS
229       CLEN=SQRT(CLENSQ)
230   C
231       IOLD=0
232       DO 70 I3=1,NWLEN
233           WLEN=WLEN(I3)
234   C K0 and the relative dielectric constant can be computed when wlen is known
235       ER=ERCOMP(WLEN)
236       K0=2*PI/WLEN
237       K0SQ=K0*K0
238       write(8,*)'Relative dielectric constant ',ER
239   C
240       DO 80 I4=1,NANG
241           ANG=ANG(I4)
242   C Theta is the angle between the incident direction and the normal to the
243   C reference plane in radians
244       THETA=PI*ANG/180.0
245       CSTHT=COS(THETA)
246       SNTHT=SIN(THETA)
247   C
248   C write(8,*)'Angle(deg) Wlen Mean Sq Slope Mean Sq Hgt'
249       write(8,*)ANG,WLEN,SIGS,HMSQ
250   C
251   C
252   C THIS SECTION COMPUTES THE SCALAR Q VALUES FOR 3 AUTO-CORR FUNCTIONS
253   C RCODE=1 -> GAUSSIAN RCODE=2 -> N=8 RCODE=3 -> N=6
254   C IF ALL Q VALUES ARE TREATED AS 0 THEN IFLAG1=0 AND THE IDD'S DO NOT
255   C NEED TO BE CALCULATED.
256       IFLAG1=0
257       DO 88 RCODE=1,3
258           CALL QCMP(Q(RCODE),AERR1,RERR1,RCODE)
259           IF(Q(RCODE).GT.0.)IFLAG1=1
260       88 CONTINUE
261   C
262   C
263   C THIS SECTION COMPUTES THE 16 IDD VALUES NEEDED FOR THE MUELLER MTX.
264   C NOTE ONLY 5 DISTINCT INTEGRATIONS ARE DONE. THE OTHERS ARE KNOWN FOR
265   C OTHER REASONS ASSIGNED TO THE FOLLOWING CONDITIONAL ASSIGNMENTS.
266   C IF IFLAG1=1 THEN THERE IS A REASON TO CALCULATE THESE VALUES
267   C IF ER IS HELD CONSTANT OVER A RANGE OF WAVELENGTHS THEN IDD'S ONLY
268   C NEED TO BE CALCULATED ONCE FOR EACH INCIDENT ANGLE.

```

# Appendix V

```

269      IF(FLAG1.GT.0.) THEN
270      IF(DCODE.NE.1.OR.IOLD.NE.1) THEN
271      DO 90 IL=1,16
272      IF(IL.EQ.1.OR.IL.EQ.2.OR.IL.EQ.4.OR.IL.EQ.5.OR.IL.EQ.6)THEN
273      CALL IDDCMP(IDD(IL),AERR2,RERR2,IL)
274      ELSE IF(IL.EQ.3.OR.IL.EQ.7) THEN
275      IDD(IL)=IDD(2)
276      ELSE IF(IL.EQ.8) THEN
277      IDD(IL)=0.
278      ELSE IF(IL.EQ.9) THEN
279      IDD(IL)=0.
280      ENDIF
281      90  CONTINUE
282      C IF Er is constant then save IDD's for next wavelength
283      IF(DCODE.EQ.1) THEN
284      DO 93 IL=1,16
285      SVIDD(4,IL)=IDD(IL)
286      93  CONTINUE
287      ENDP
288      ELSE
289      C IF Er is constant and the IDD's have been saved use them
290      DO 92 IL=1,16
291      IDD(IL)=SVIDD(4,IL)
292      92  CONTINUE
293      ENDP
294      ENDP
295      C
296      C
297      C THIS SECTION UTILIZED THE IDD VALUES TO COMPUTE THE MUELLER MTX
298      C ELEMENTS DIVIDED BY 1.
299      C P(1)=P11/Q, P(2)=P12/Q, ... P(5)=P21/Q, ... P(16)=P44/Q
300      C MANY OF THESE P VALUES ARE ZERO BUT THEY ARE CALCULATED HERE FOR
301      C COMPLETENESS. THEIR CALCULATION TIME IS MINUTE COMPARED TO THE ACTUAL
302      C INTEGRATIONS.
303      P(1)=.5*(IDD(1)+IDD(2)+IDD(3)+IDD(4))
304      P(2)=.5*(IDD(1)+IDD(2)-IDD(3)-IDD(4))
305      P(3)=IDD(9)
306      P(4)=-IDD(10)
307      P(5)=.5*(IDD(1)-IDD(2)+IDD(3)-IDD(4))
308      P(6)=.5*(IDD(1)-IDD(2)-IDD(3)+IDD(4))
309      P(7)=IDD(13)
310      P(8)=-IDD(14)
311      P(9)=2.*IDD(11)
312      P(10)=2.*IDD(15)
313      P(11)=IDD(5)+IDD(7)
314      P(12)=-IDD(6)+IDD(8)
315      P(13)=2.*IDD(12)
316      P(14)=2.*IDD(16)
317      P(15)=IDD(6)*IDD(8)
318      P(16)=IDD(5)-IDD(7)
319      C
320      C
321      C THE SCALAR Q VALUES ARE WRITTEN ON ONE LINE AND
322      C 6 MUELLER MATRIX ELEMENTS DIVIDED BY Q ARE WRITTEN ON THE NEXT.

```

# Appendix V

```

323 C THE OTHER 10 ARE KNOWN TO BE ZERO FOR THIS WORK.
324 WRITE(9,2000)Q(1),Q(2),Q(3)
325 WRITE(9,2000)P(1),P(2),P(6),P(11),P(12),P(16)
326 2000 FORMAT(6E12.4)
327 write(8,2001)Q(1),Q(2),Q(3)
328 write(8,2001)IDD(1),IDD(2),IDD(4),IDD(5),IDD(6)
329 2001 FORMAT(6E12.4)
330 C
331 80 CONTINUE
332 IOLD=1
333 70 CONTINUE
334 60 CONTINUE
335 50 CONTINUE
336 C
337 REWIND(7)
338 REWIND(8)
339 REWIND(9)
340 CLOSE(7)
341 CLOSE(8)
342 CLOSE(9)
343 STOP
344 END
345 C
346 C
347 C
348 C THIS SUBROUTINE DRIVES QDAG TO COMPUTE Q
349 SUBROUTINE QCMP(Q,AERR,RERR,RCODE)
350 REAL VX,VY,VXZ,VYSQH,UPLIM,QPRIME,SETARG,ARG,ARG1,
351 1 UPLIMD,UPLIM1,SETUPR
352 REAL HMSQ,CLEN,CLENSQ,WLEN,THETA,SNHT,CSHT,SIGS,K0,PI
353 REAL GCNTI
354 INTEGER RCODE
355 EXTERNAL ARG
356 COMMON/ONE/HMSQ,CLEN,CLENSQ,WLEN,THETA,SNHT,CSHT,SIGS,K0,PI
357 C FOR EACH VALUE OF THETA COMPUTE COMPONENTS OF VECTOR V.
358 C THEN COMBINE WITH THE MEAN SQUARE HEIGHT AND SAVE IN ARG
359 C WRITE(8,'YSNHT,CSHT,SNHT,CSHT
360 VX=-2.*K0*SNHT
361 C VZ=0.
362 VY=2*K0*CSHT
363 VXZ=ABS(VX)
364 VYSQH=VY*VY*HMSQ
365 SETARG=ARG1(VXZ,VYSQH)
366 UPLIMD=SETUPR(RCODE)
367 SETARG=GCNTI(0.)
368 C COMPUTE THE UPPERLIMIT ON THE INTEGRATION BY ASSUMING THE INTEGRAND
369 C DIES AWAY WITHIN UPLIMD CORRELATION LENGTHS.
370 UPLIM=UPLIMD*CLEN
371 UPLIM1=UPLIM/100.
372 100 CONTINUE
373 C COMPUTE THE DOUBLE INTEGRAL WHERE ONE HALF -> 2PI
374 C THIS CAN BE DONE BY SWITCHING TO POLAR COORDINATES
375 C IF NO INTERGRAND IS FOUND THEN REDUCE INTEGRATION LIMITS TO FIND IT
376 CALL QDAG(ARG,0.,UPLIM,AERR,RERR,1,QPRIME,ERREST)

```

# Appendix V

```

377      QPRIME=QPRIME**2*PI
378      IF(QPRIME.EQ.0.AND.UPLIM.GT.UPLIM1) THEN
379          UPLIM=UPLIM*0.7
380          GOTO 100
381      ENDIF
382  C
383  C IF QPRIME > 0. THEN COMPUTE THE TOTAL Q
384  C IF QPRIME STILL EQUALS ZERO THEN INDICATE BY -998.
385  C IF QPRIME WAS LESS THAN ZERO INDICATE BY -999.
386      IF (QPRIME.GT.0.)THEN
387          Q=(K0**K0/PI)*QPRIME
388      ELSE IF(QPRIME.EQ.0.)THEN
389          Q=-998.
390      ELSE IF(QPRIME.LT.0.)THEN
391          Q=-999.
392      ENDIF
393  C  WRITE(8,'Q-',Q,'  ACCESSED',GCNT1(0.)
394      RETURN
395      END
396  C
397  C
398  C
399  C THIS FUNCTION COMPUTES THE INTEGRAND OF Q FOR DCADRE.
400  C THIS FUNCTION HAS 3 ENTRY POINTS
401  C BSJ0 COMPUTES BESSEL FUNCTION J SUB 0
402      FUNCTION ARG(RD)
403      REAL BSJ0,RRSURF,CHT2,CHTSQ,JSUB0,RD,VXZ,VYSQH
404      REAL COUNT,DUMMY
405      INTEGER RCODE,DRCODE
406      SAVE
407      COUNT=COUNT+1.
408      JSUB0=BSJ0(VXZ*RD)
409      CHT2=EXP((RRSURF(RD)-1.)*VYSQH)
410      ARG=JSUB0*(CHT2-CHTSQ)*RD
411  C  ARG-1.
412      RETURN
413  C
414  C
415  C THIS SECOND ENTRY POINT SAVES SOME CONSTANTS FOR A GIVEN INTEGRATION
416  C THESE VALUES ARE ONLY FUNCTIONS OF THE SURFACE PARAMETERS NOT OF RD
417      ENTRY ARG1(DVXZ,DVYSQH)
418      VXZ=DVXZ
419      VYSQH=DVYSQH
420      CHTSQ=EXP(-VYSQH)
421      ARG1=1.0
422      RETURN
423  C
424  C
425  C THIS ENTRY RETURNS THE NUMBER OF ACCESSES SINCE LAST INQUIRY
426      ENTRY GCNT1(DUMMY)
427      GCNT1=COUNT
428      COUNT=DUMMY
429      RETURN
430      END

```

# Appendix V

```

431 C
432 C
433 C THIS FUNCTION CALCULATES THE AUTOCORRELATION FUNCTION USED TO
434 C MODEL THE RANDOMLY ROUGH SURFACE
435 C BSK1 COMPUTES THE BESSEL FUNCTION K SUB 1
436 C BSK0 COMPUTES THE BESSEL FUNCTION K SUB 0
437 FUNCTION RRSURF(RD)
438 REAL RD,R,RSQ,R4TH,R6TH,KAPPA6,KAPPA8,BSK1,BSK0,TERM1,TERM2
439 REAL HMSQ,CLEN,CLENSQ,WLEN,THETA,SNHTT,CSTHT,SIGS,K0,PI
440 INTEGER RCODE,DRCODE
441 COMMON/ONE/HMSQ,CLEN,CLENSQ,WLEN,THETA,SNHTT,CSTHT,SIGS,K0,PI
442 IF(RD.EQ.0.) THEN
443   RRSURF=1.
444   RETURN
445 ENDIF
446 IF(RCODE.EQ.1) THEN
447   RRSURF=EXP(-RD*RD/CLENSQ)
448   RETURN
449 ELSE IF(RCODE.EQ.2) THEN
450   R=RD*KAPPA8
451   RSQ=R*R
452   R4TH=RSQ*R4TH
453   R6TH=RSQ*R6TH
454   TERM1=(1.-3.*RSQ/8.+3.*R4TH/32.+R6TH/3072.)*R*BSK1(R)
455   TERM2=(RSQ/2.-R4TH/4.-R6TH/96.)*BSK0(R)
456   RRSURF=TERM1+TERM2
457   RETURN
458 ELSE IF(RCODE.EQ.3) THEN
459   R=RD*KAPPA6
460   RSQ=R*R
461   R4TH=RSQ*R4TH
462   TERM1=(1.-3.*RSQ/4.-R4TH/96.)*R*BSK1(R)
463   TERM2=(RSQ/2.+3.*R4TH/16.)*BSK0(R)
464   RRSURF=TERM1+TERM2
465   RETURN
466 ELSE
467   WRITE(8,*)'ILLEGAL RCODE'
468   STOP
469 ENDIF
470 RRSURF=0.
471 RETURN
472 C
473 C
474 ENTRY SETUPR(DRCODE)
475 RCODE=DRCODE
476 SETUPR=1.
477 IF(RCODE.EQ.1) THEN
478   SETUPR=5.
479 ELSE IF(RCODE.EQ.2) THEN
480   KAPPA8=SQRT(1.6)/CLEN
481   SETUPR=175./SQRT(1.6)
482 ELSE IF(RCODE.EQ.3) THEN
483   KAPPA6=1./CLEN
484   SETUPR=175.

```

# Appendix V

```

485     ENDIF
486     RETURN
487     END
488 C
489 C
490 C
491 C THIS SUBROUTINE DRIVES TWODQ TO COMPUTE IDD FOR DI*DKL IN INTEGRAND
492 C THE REAL PART IS COMPUTED FOR CODE= 1 OR 2, IMAGINARY CODE= 3
493     SUBROUTINE IDDCMP(IDD,AERR,RERR,IL)
494     REAL IDD,AERR,RERR,ERREST
495     REAL HZMIN,HZMAX,HXMIN,HXMAX,HZMAX1
496     REAL SDPQ,SP2,SPHXHZ,SZMIN,SZMAX
497     REAL HMSQ,CLEN,CLENSQ,WLEN,THETA,SNHTT,CSTHT,SIGS,K0,PI
498     REAL GCNT2
499     INTEGER ICODE(16),ISUB(16),JSUB(16),KSUB(16),LSUB(16)
500     INTEGER IL
501     COMPLEX ERCMP,UR
502     EXTERNAL ARGIDD,ZMIN,ZMAX
503     COMMON/ONE/HMSQ,CLEN,CLENSQ,WLEN,THETA,SNHTT,CSTHT,SIGS,K0,PI
504     DATA ICODE/1,1,1,1,2,3,2,3,2,3,2,3,2,3,2,3/
505     DATA ISUB/1,2,2,1,1,1,1,1,1,1,1,2,2,1,1,1/
506     DATA JSUB/1,2,1,2,1,1,2,2,1,1,1,1,1,1,2,2/
507     DATA KSUB/1,1,2,2,2,2,2,1,1,2,2,2,2,2,2/
508     DATA LSUB/1,2,1,2,2,2,1,1,2,2,1,1,2,2,2,2/
509 C
510     UR=(1.,0.)
511     SETUP=SDPQ(ICODE(IL),ISUB(IL),JSUB(IL),KSUB(IL),
512     1 LSUB(IL),ERCMP(WLEN),UR,CSTHT,SNHTT)
513     SETUP=SP2(SIGS,SNHTT,CSTHT,PI)
514 C WRITE(8,*)'FACT= ',SETUP
515     SETUP=SPHXHZ(PI,SIGS)
516 C WRITE(8,*)'DENOM= ',SETUP
517     SETUP=GCNT2(0.)
518     HZMIN=0.
519     HZMAX=3.*SQRT(SIGS)
520     HZMAX1=HZMAX*.4
521     HXMIN=HZMAX
522     HXMAX=HZMAX
523     199 SETUP=SZMIN(HZMIN)
524     SETUP=SZMAX(HZMAX)
525 C
526     CALL TWODQ(ARGIDD,HXMIN,HXMAX,ZMIN,ZMAX,AERR,RERR,1,IDD,ERREST)
527     IDD=2.*IDD
528 C
529 C WRITE(8,*)'IDD= ',IDD,' ACCESSED',GCNT2(0.)
530     IF (IDD.EQ.0.0.AND.HZMAX.GT.HZMAX1)THEN
531         HZMIN=HZMIN*.7
532         HZMAX=HZMAX*.7
533         HXMIN=HXMIN*.7
534         HXMAX=HXMAX*.7
535         GOTO 199
536     ENDIF
537 C
538     RETURN

```

# Appendix V

```

539      END
540      C
541      C
542      C
543      C THESE FUNCTIONS CALCULATE THE HZ LIMITS FOR 2-D INTEGRATION BY TWODQ
544      FUNCTION ZMIN(X1)
545      REAL D1ZMIN,D2ZMIN
546      SAVE
547      ZMIN=D1ZMIN
548      RETURN
549      ENTRY SZMIN(D2ZMIN)
550      D1ZMIN=D2ZMIN
551      SZMIN=1.0
552      RETURN
553      END
554      C
555      FUNCTION ZMAX(X1)
556      REAL D1ZMAX,D2ZMAX
557      SAVE
558      ZMAX=D1ZMAX
559      RETURN
560      ENTRY SZMAX(D2ZMAX)
561      D1ZMAX=D2ZMAX
562      SZMAX=1.0
563      RETURN
564      END
565      C
566      C
567      C
568      C THIS FUNCTION CALCULATES THE ARGUMENT OF IDD FOR TWODQ
569      FUNCTION ARGIDD(HX,HZ)
570      REAL COUNT,DUMMY,HXZSQ
571      REAL DENOM,DSIGS,SIGS,SNHTT,CSTHT,P1,FACT
572      REAL SIGSRT,PIC,C0,TANT,COTT,R,P1,P2
573      REAL ERPC
574      SAVE
575      C
576      COUNT=COUNT+1.
577      HXZSQ=HX*HX+HZ*HZ
578      C
579      IF (HX.LT.-COTT) THEN
580          P2=0.
581      ELSE
582          P2=FACT
583     ENDIF
584      C
585      PHXHZ=EXP(-HXZSQ/SIGS)/DENOM
586      C
587      ARGIDD=PHXHZ*P2*DPQMAG(HX,HZ,HXZSQ)
588      RETURN
589      C
590      C
591      ENTRY SPHXHZ(P1,DSIGS)
592      SIGS=DSIGS

```

# Appendix V

```

593      DENOM=PPSIGS
594      SPHXHZ=DENOM
595      RETURN
596 C
597 C
598      ENTRY SP2(DSIG5,SNTHT,CSHTT,P1)
599      SIGSRT=SQRT(DSIG5)
600      PIC=.5*SIGSRT/SQRT(P1)
601      IF(ABS(SNTHT).LT.1.E-10)THEN
602          COTT=1.E30
603          CO=0.
604      ELSE
605          TANT=SNTHT/CSHTT
606          COTT=1/TANT
607          R=COTT/SIGSRT
608          P1=EXP(-R*R)
609          P2=ERFC(R)
610          CO=PIC*TANT*P1-.5*P2
611      ENDP
612      FACT=1./(1.+CO)
613      SP2=FACT
614      RETURN
615 C
616 C
617      ENTRY GCNT2(DUMMY)
618      GCNT2=COUNT
619      COUNT=DUMMY
620      RETURN
621      END
622 C
623 C
624 C
625 C THIS FUNCTION CALCULATE DIJ*DKL AS PART OF IDD'S INTEGRAND
626      FUNCTION DPQMAG(HX,HZ,HXZSQ)
627      COMPLEX ER,UR,RIR,ETAR,ERM,ERM,ERMRR,URM,URMR,URMRR
628      COMPLEX SN1N,CS1N,DEN2,DEN3,C1P,CPVV,CPHH,B1,B2,B3,B4
629      COMPLEX DPQ(2,2),DER,DUR
630      REAL HX,HZ,HXZSQ,SNTHT,CSHTT,DSNTHT,DCSTHT
631      REAL CSG,TNG,SNG,CS0N,SN0N,SNPDI,CSPDI,SOP,COIP
632      REAL CSSI,SN5II,SN5IP
633      INTEGER ICODE,ICODE,ISUB,ISUB,JSUB,JSUB,KSUB,KSUB,LSUB,LLSUB
634      SAVE
635 C
636      CSG=1./SQRT(1.+HXZSQ)
637      TNG=SQRT(HXZSQ)
638      SNG=CSG*TNG
639      IF(TNG.LT.1.E-5) THEN
640          CS0N=CSHTT
641          SN0N=SNTHT
642      ELSE
643          CSPDI=-HX/TNG
644          SNPDI=-HZ/TNG
645          CS0N=CSG*CSHTT-SNG*SNTHT*CSPDI
646          SN0N=SQRT(1.-CS0N*CS0N)

```

# Appendix V

```

647     ENDF
648 C
649     IF(ABS(CSON).LT.1.E-5) THEN
650         DPQMAG=0.
651         RETURN
652     ENDF
653 C
654     SN1N=SNON/RUR
655     CS1N=CSQRT(1.-SN1N*SN1N)
656     DEN2=CSON+CS1N*ETAR
657     DEN3=CSON+CS1N/ETAR
658     S0IF=SNON*SNON
659     C0IF=CSON
660     C1IF=CS1N*CS1N
661     CPVV=COIF*((-UR*C1IF-S0IF)*ERMUR-URM)/(DEN2*DEN2)
662     CPHH=COIF*((-ER*C1IF-S0IF)*URMUR-ERM)/(DEN3*DEN3)
663 C
664     IF(TNG.LT.1.E-5) THEN
665         DPQ(1,1)=CPVV
666         DPQ(1,2)=0.
667         DPQ(2,1)=0.
668         DPQ(2,2)=CPHH
669     ELSE
670         CSSI=(CSG*SNTHT+SNG*CSHT*CSPI)/SNON
671         SNSII=-SNG*SNPI/SNON
672         SNSIP=-SNSII
673         B1=CPVV*CSSI
674         B2=-CPHH*SNSII
675         B3=CPVV*SNSII
676         B4=CPHH*CSSI
677         DPQ(1,1)=CSSI*B1-SNSIP*B2
678         DPQ(1,2)=CSSI*B3-SNSIP*B4
679         DPQ(2,1)=SNSIP*B1+CSSI*B2
680         DPQ(2,2)=SNSIP*B3+CSSI*B4
681     ENDF
682 C
683     IF(ICODE.EQ.1) THEN
684         DPQMAG=(CABS(DPQ(ISUB,JSUB))/CSG)**2
685     ELSE IF(ICODE.EQ.2) THEN
686         DPQMAG=REAL(DPQ(ISUB,JSUB)*CONJG(DPQ(KSUB,LSUB)))/CSG/CSG
687     ELSE
688         DPQMAG=AIMAG(DPQ(ISUB,JSUB)*CONJG(DPQ(KSUB,LSUB)))/CSG/CSG
689     ENDF
690     RETURN
691 C
692 C
693     ENTRY SDPQ(ICODE,ISUB,JSUB,KSUB,LSUB,DER,DUR,
694 1     DCSTHT,DSNTHT)
695     ICODE=ICODE
696     ISUB=ISUB
697     JSUB=JSUB
698     KSUB=KSUB
699     LSUB=LSUB
700     CSTHT=DCSTHT

```

# Appendix V

```

701      SNTHT-DSNTHT
702      ER-DEB
703      UR-DUR
704      RIR-CSQRT(ER*UR)
705      ETAR-CSQRT(UR/ER)
706      ERM-1.-ER
707      ERMR-1.-1./ER
708      ERMRR-ERMRR*RIR
709      URM-1.-UR
710      URMRR-1.-1./UR
711      URMRR-URMR*RIR
712      SDPQ-1.0
713      RETURN
714      END
715  C
716  C
717  C
718  C THIS FUNCTION COMPUTES THE VALUE ER IF INTERPOLATION IS DESIRED
719  C OTHERWISE IT JUST RETURNS THE CONSTANT VALUE GIVEN IN PARAMS.
720      COMPLEX FUNCTION ERCMP(WLEN)
721      INTEGER NPTS,DEG,I,J,MIN,MAX,DDEG,DNPTS
722      REAL WN(200)
723      REAL WLEN,WNP,FACTOR,LI,NR,KR
724      COMPLEX D1ER(200),EREST
725      CHARACTER*10 PNAME
726      SAVE
727  C
728      IF(NPTS.EQ.1)THEN
729          ERCMP=D1ER(1)
730          RETURN
731      ELSE
732          C this converts wavelength in microns to wave number in 1/cm
733          WNP=10000./WLEN
734          I=(DEG+1)/2
735          IF(WN(I).GT.WNP) THEN
736              MIN=I
737              MAX=MIN+DEG
738              GOTO 211
739          ENDP
740          200 IF(WN(I).GT.WNP) GOTO 210
741              I=I+1
742          IF(I.EQ.NPTS-(DEG+1)/2)THEN
743              MAX=NPTS
744              MIN=NPTS-DEG
745              GOTO 211
746          ENDP
747          GOTO 200
748      C
749          210 MIN=I-DEG/2-1
750          MAX=MIN+DEG
751          211 FACTOR = 1.
752      C  WRITE(6,"MIN,MAX ",MIN,MAX
753      C
754          DO 220 J= MIN,MAX

```

# Appendix V

```

735      IF(WNP.NE.WN(I))GOTO 220
736      ERCMP=DIER(I)
737      RETURN
738      220 FACTOR=FACTOR*(WNP-WN(I))
739      C
740      EREST=(0.,0.)
741      DO 230 I=MIN,MAX
742      LI=FACTOR*(WNP-WN(I))
743      DO 240 J= MIN,MAX
744      240 IF(I.NE.J)LI=LI/(WN(I)-WN(J))
745      230 EREST=EREST+DIER(I)*LI
746      ENDIF
747      C
748      ERCMP=EREST
749      RETURN
750      C
751      C
752      ENTRY SETER(DNPTS,DDEG)
753      NPTS=DNPTS
754      DEG=DDEG
755      IF(NPTS.EQ.1)THEN
756      READ(7,*)DIER(1)
757      ELSE
758      READ(7,')(A)PNAME
759      WRITE(8,*)OPENING 'PNAME' AS DIELECTRIC FILE
760      OPEN(UNIT=10,FILE=PNAME)
761      READ(10,*)NPTS
762      DO 277 I=1,NPTS
763      READ(10,*)WN(I),NR,KR
764      DIER(I)=(1+.0.*((NR*NR-KR*KR)-(0..1.*ABS(2.*NR*KR)
765      277 CONTINUE
766      CLOSE(10)
767      ENDIF
768      SETER=1.
769      RETURN
770      END

```

## Appendix V

### AV.6 SAMPLE CALCULATIONS: PARAMETERS INPUT, COMMENTARY AND DATA OUTPUT FILES.

"Params" is an input file for RETRO, containing topographical parameters about the scatterer and routine control information. UNIX output files "TestC1c.d" and "TestC1c.c" are created by RETRO for calculation inspections and trouble-shooting purposes. (1) "Test" implies the file was created by RETRO, and is not a combination of other output files, (2) upper case "C" implies that this is a composite clay material, (3) "0" and lower case "c" imply that mean-squared height and slope of the clay surface is  $5 \mu m^2$  and 0.5 respectively, and ".c" or ".d" at the end indicates commentary file or data file for DISPLAY, respectively.

#### AV.6.1 Sample data file "PARAMS." (See Appendix V.1.1)

line # (No leading spaces in first column)

```

1 TestC1c.c
2 TestC1c.d
3 composite clay
4 -1 0. 0.
5 1.5
6 -1 0. 0.
7 .5
8 71 9. 0.05
9 22 0. 4.
10 0
11 compos.nk
12 .000000001 .0000001
13 .00001 .005
```

#### AV.6.2 Sample data file TestC1c.d. (See Appendix AV.1.4)

```

1. composite clay
2. 0001 0001 0071 0022
3. 0.1500E+01
4. 0.5000E+00
5. 0.9000E+01 0.9050E+01 0.9100E+01 0.9150E+01 0.9200E+01
6. 0.9250E+01 0.9300E+01 0.9350E+01 0.9400E+01 0.9450E+01
7. 0.9500E+01 0.9550E+01 0.9600E+01 0.9650E+01 0.9700E+01
8. 0.9750E+01 0.9800E+01 0.9850E+01 0.9900E+01 0.9950E+01
9. 0.1000E+02 0.1005E+02 0.1010E+02 0.1015E+02 0.1020E+02
10. 0.1025E+02 0.1030E+02 0.1035E+02 0.1040E+02 0.1045E+02
11. 0.1050E+02 0.1055E+02 0.1060E+02 0.1065E+02 0.1070E+02
12. 0.1075E+02 0.1080E+02 0.1085E+02 0.1090E+02 0.1095E+02
13. 0.1100E+02 0.1105E+02 0.1110E+02 0.1115E+02 0.1120E+02
14. 0.1125E+02 0.1130E+02 0.1135E+02 0.1140E+02 0.1145E+02
15. 0.1150E+02 0.1155E+02 0.1160E+02 0.1165E+02 0.1170E+02
16. 0.1175E+02 0.1180E+02 0.1185E+02 0.1190E+02 0.1195E+02
17. 0.1200E+02 0.1205E+02 0.1210E+02 0.1215E+02 0.1220E+02
18. 0.1225E+02 0.1230E+02 0.1235E+02 0.1240E+02 0.1245E+02
19. 0.1250E+02
20. 0.0000E+00 0.4000E+01 0.8000E+01 0.1200E+02 0.1600E+02
21. 0.2000E+02 0.2400E+02 0.2800E+02 0.3200E+02 0.3600E+02
22. 0.4000E+02 0.4400E+02 0.4800E+02 0.5200E+02 0.5600E+02
```

## Appendix V

23. 0.6000E+02 0.6400E+02 0.6800E+02 0.7200E+02 0.7600E+02  
24. 0.8000E+02 0.8400E+02  
25. 0.2446E+01 0.1649E+01 0.1901E+01  
26. 0.2483E+00 -0.1437E-11 0.2336E+00 -0.2188E+00 -0.1539E-12 -0.2336E+00  
27. 0.2412E+01 0.1605E+01 0.1896E+01  
28. 0.2482E+00 0.3166E-03 0.2332E+00 -0.2181E+00 -0.5693E-03 -0.2332E+00  
29. 0.2312E+01 0.1703E+01 0.2447E+01  
30. 0.2478E+00 0.1243E-02 0.2320E+00 -0.2162E+00 -0.2228E-02 -0.2320E+00

3139.	0.3237E-02	0.4312E-02	0.8359E-02			
3140.	0.6718E-01	0.1580E-01	0.6122E-01	-0.5657E-01	-0.4200E-02	-0.6252E-01
3141.	0.5769E-03	0.1179E-02	0.3512E-02			
3142.	0.5872E-01	0.1476E-01	0.5334E-01	-0.4907E-01	-0.3925E-02	-0.5445E-01
3143.	0.6616E-04	0.3242E-03	0.1474E-02			
3144.	0.4887E-01	0.1313E-01	0.4427E-01	-0.4053E-01	-0.3492E-02	-0.4513E-01
3145.	0.4011E-05	0.9443E-04	0.5812E-03			
3146.	0.3741E-01	0.1075E-01	0.3380E-01	-0.3074E-01	-0.2857E-02	-0.3435E-01
3147.	0.7976E-07	0.2454E-04	0.1781E-03			
3148.	0.2409E-01	0.7396E-02	0.2172E-01	-0.1961E-01	-0.1965E-02	-0.2198E-01

**AV.6.3 Commentary file TestC1c.c. (See Appendix AV.1.3)**

- ```

1  Commentary for retro
2  Output file is  TestC1c.d
3  Material description is  Composite Clay
4  1,    Mean square height values read in
5  1,    Mean square slope values read in
6  71,   Wavelengths read in
7  22,   Incident angles read in
8  Dielectric code read in  0
9  Opening compos.nk  as dielectric file
10 The following relative dielectrics are used
11 wlen,Er 9.0, (0.230451874,-1.11635916)
12 wlen,Er 9.05, (0.127786514,-0.9956851)
13 wlen,Er 9.1, (-0.12072582,-0.933739048)
14 wlen,Er 9.15, (-0.349295173,-1.01144447)
15 wlen,Er 9.2, (-0.580000781,-1.14418849)
16 wlen,Er 9.25, (-0.813025525,-1.31932772)
17 wlen,Er 9.3, (-1.06799906,-1.5584366)
18 wlen,Er 9.35, (-1.33129798,-1.88047748)
19 wlen,Er 9.4, (-1.59029517,-2.30134942)
20 wlen,Er 9.45, (-1.82918509,-2.86734854)
21 wlen,Er 9.5, (-2.00403183,-3.61813835)
22 wlen,Er 9.55, (-2.00863271,-4.69301692)
23 wlen,Er 9.6, (-1.62000141,-5.93069545)
24 wlen,Er 9.65, (-7.286277419E-02,-6.88245115)

```

# Appendix V

25 wlen,Er 9.7, (1.9116751,-7.26745517)  
 26 wlen,Er 9.75, (2.69681864,-6.34696569)

.  
 .  
 .  
 .  
 .  
 .  
 .

77 wlen,Er 12.3, (2.19544571,-0.322365349)  
 78 wlen,Er 12.35, (2.14063373,-0.355664223)  
 79 wlen,Er 12.4, (2.1129991,-0.433485636)  
 80 wlen,Er 12.45, (2.10166741,-0.515307218)  
 81 wlen,Er 12.5, (2.106225,-0.586)  
 82 Error criteria for QDAG read in 1.0E-09, 1.0E-07  
 83 error criteria for TWODQ read in 1.0E-05, 5.0E-03

84 Cray Research's UNICOS Release 3.0.2

86  
 87 /amsaa is full, please clean up.  
 88 /secad is full, please clean up.

89  
 90 Input TERM type? Tek 4115

91  
 92 COMMAND REPORT

93  
 94 COMMAND STARTED USER-CPU SYS-CPU I/O-WAIT ELAPSED SBU'S  
 95 NAME AT [SECONDS] [SECONDS] [SECONDS] [SECONDS]

96  
 97 jad 10:04:49 0.01 0.03 0.00 0.09 0.00  
 98 segldr 10:04:50 2.94 1.29 0.04 10.95 0.00  
 99 sh 10:04:49 0.08 0.07 0.00 3556.54 0.00  
 100 jad 10:04:50 0.10 0.97 0.00 3556.13 0.00  
 101 a.out 10:05:01 1832.13 6.60 0.00 3545.12 0.00  
 102 a.out 15:08:32 1562.01 2.51 0.00 2457.13 0.00

103  
 104 PROCESS FLOW CHART

105 -----

106  
 107 parent -> child ...

108 -----

109  
 110 jad  
 111 segldr  
 112 sh  
 113 jad  
 114 a.out  
 115 a.out

116  
 117 JOB ACCOUNTING REPORT

118 -----

119 Operating System : XXXX  
 120 User : XXXX

# Appendix V

|     |                            |   |                         |
|-----|----------------------------|---|-------------------------|
| 121 | Group                      | : | XXXX                    |
| 122 | Accounting Id              | : | XXXX                    |
| 123 | Job Id                     | : | XXXX                    |
| 124 | Report starts              | : | 07/14/88 10:04:49       |
| 125 | Report ends                | : | 07/14/88 15:49:29       |
| 126 | CPU Time (User)            | : | 3397.2720 Seconds       |
| 127 | CPU Time (System)          | : | 11.4611 Seconds         |
| 128 | I/O Wait Time              | : | 0.0415 Seconds          |
| 129 | Elapsed Time               | : | 20680 Seconds           |
| 130 | CPU Time Memory Integral   | : | 0.8662 MWords * Seconds |
| 131 | I/O W-Time Memory Integral | : | 0.1419 MWords * Seconds |
| 132 | Data Transferred           | : | 17.0252 MBytes          |
| 133 | Logical I/O Requests       | : | 6454                    |
| 134 | Real I/O Requests          | : | 142                     |
| 135 | No. of Commands            | : | 6                       |
| 136 | Billing Units              | : | 0.0000                  |

#### AV.7.1 DISPLAY User's Guide: STARTUP PROCEDURE.

To run DISPLAY, log on and type "display.run" after the prompt. This runstream compiles, links, and executes the program. DISPLAY's executable code is stored in the file display.xqt. If one desires to rerun DISPLAY during the same login session, type "retro.xqt." This will run the program again without unnecessary recompiling and relinking. Since retro.xqt is a very large file, it should be removed before logging off. The object file, display.o, is removed immediately following the linking process.

At the beginning of each run, the program will ask for the graphics terminal type. DISPLAY directly supports the Tektronics 4107 and 4115 device drivers. After the terminal type is defined, DISPLAY enters its main menu.

A note about Tektronics terminals. If the page border and words of a DISPLAY plot are drawn in dashed lines, the problem is that the CRAY has changed the terminal setup status. A solution to this problem is simply wait for the plot to finish then press the reset button on the terminal. This will clear the screen, self test, and reset the terminal's setup parameters. When the self test is finished, press <RETURN> to continue.

#### AV.7.2 MAIN MENU OPTIONS.

After the menu described below has been printed, the program prompts your input. As with all menu prompts in this program, type in the number of your choice and hit <RETURN>. If you type an illegal value, the menu will be reprinted and you should try again. After the main menu option is entered, additional menu prompts will guide you. If you make a mistake and do not want any of the options presented at a given menu, try typing in "0" to back up a menu.

The main menu has 13 options, given below.

##### DISPLAY MAIN MENU

- 11. Assign a surface for data access
- 12. Display info. on an assigned surface
- 13. Remove an assigned surface
- 21. Modify list of surfaces to use
- 22. Modify list of matrix elements to use
- 23. Modify list of wavelengths to use
- 24. Modify list of incident angles to use
- 31. Change analysis normalizations
- 32. Change plotting format features
- 41. Do 3-d plot
- 42. Do 2-d plot
- 43. Do Contour plot
- 99. Quit program

## Appendix V

The first three options 11, 12 and 13 are grouped as the data file control section. Option 11 acquires the file name, auto-correlation function, mean squared height ( $\langle h^2 \rangle$ ), and mean squared slope ( $\sigma_f^2$ ) that defines a surface. It then assigns a number to that sample. Option 12 displays pertinent information on the assigned surfaces, thus you need not remember or write down all the assignment numbers. Option 13 allows one to remove a surface assignment.

The next four options, 21, 22, 23, and 24 maintain the directive lists used by the plotting routines. Option 21 builds a surface assignment list for use by three plotting options, determining which surface(s) to use for a particular plot. Option 22 maintains the list of matrix elements to use when plotting 2-d representations of the data. Option 23 maintains the list of fixed  $\lambda_0$ 's for use when plotting 2-d angle relationships of the Mueller elements. Lastly, Option 24 maintains the list of fixed  $\theta_0$ 's used in plotting the matrix elements as a function of  $\lambda_0$ .

Option 31 controls which analysis operations are used when data is read.

In addition to  $f_{ij}$ , it is also possible to plot  $\frac{f_{ij}}{f_{11}}, \frac{f_{ij}^{lrg}}{f_{ij}^{base}} - 1$ , or  $\frac{f_{ij}^{lrg}}{f_{11}^{lrg}} \frac{f_{11}^{base}}{f_{ij}^{base}} - 1$ .

Option 32 allows modification of plot features such as page size, line color, tickmark density, grid lines, curve thickness, and hidden lines on 3-d plots.

Options 41, 42, and 43 produce the actual plot after the data files have been assigned, the directive lists are built, and the plot format established. Option 41 plots, in a 3-d format, a single matrix element of a single surface as a function of  $\lambda_0$  and  $\theta_0$ . Option 42 graphs 2-d cross sections of 3-d plots with either  $\theta_0$  or  $\lambda_0$  held constant. This can be done for multiple matrix elements, one per page. Option 43 draws a contoured representation of 3-d plots. Up to four surfaces can be compared on one page.

Option 99 terminates the program. When exiting "bob>," remember to delete the large file retro.xqt.

### AV.7.3 SURFACE ASSIGNMENT OPTIONS.

These three routines connect DISPLAY to the desired data files. You must know the file names before you begin. Required information is obtained from prompted user inputs. The program works on a system of surface assignment numbers. A surface is defined by a file name, auto-correlation function (a special code is reserved for experimental data),  $\langle h^2 \rangle$ , and  $\sigma_f^2$ . Each surface has its own list of allowed  $\theta_0$ 's and  $\lambda_0$ 's stored in the heading section of the data file. Up to ten surfaces can be assigned at one time.

#### AV.7.3.1 Option 11.

Option 11 accesses the file name, auto-correlation function,  $\langle h^2 \rangle$ , and  $\sigma_f^2$  that define a surface, then assigns a number to that surface. This routine next reads in the allowed  $\lambda_0$ 's and  $\theta_0$ 's. This option does not read the Mueller matrix elements, but retains the above information and associates it with the assignment number given to the surface. The matrix elements are read, as needed, for plotting.

The data file contains two distinct sections: the heading block and the data block. The heading block is the key to finding the elements of Mueller matrix,  $F(\langle h^2 \rangle, \sigma_f^2, \lambda_0, \theta_0)$ . The format of the heading block is as follows:

- A - description of material
- B - no. of  $\langle h^2 \rangle$ 's,  $\sigma_z^2$ 's,  $\lambda_0$ 's, and  $\theta_0$ 's
- C -  $\langle h^2 \rangle(1) \dots \langle h^2 \rangle(\text{no. of } \langle h^2 \rangle\text{'s})$
- D -  $\sigma_z^2(1) \dots \langle h^2 \rangle(\text{no. of } \sigma_z^2\text{'s})$
- E -  $\lambda_0(1) \dots \lambda_0(\text{no. of } \lambda_0\text{'s})$
- F -  $\theta_0(1) \dots \theta_0(\text{no. of } \theta_0\text{'s})$

The information listed on lines C through F, above, may actually occupy several lines in the data file.

For files created by RETRO the data block consists of two data lines per matrix. The first line contains three Q values corresponding to the auto-correlation functions Gaussian, N=8, and N=6. The second line contains six generally non-zero matrix elements divided by Q. In order, these correspond to  $f_{11}$ ,  $f_{12}$ ,  $f_{22}$ ,  $f_{33}$ ,  $f_{34}$ , and  $f_{44}$ . RETRO does not write  $f_{21} = f_{12}$  or  $f_{43} = -f_{34}$ . The other 8 matrix elements are zero.

For files containing experimental data the data block consists of three data lines per matrix. The first line contains the 11, 12, 13, 14, 21, and 22 elements of F. The second line contains the 23, 24, 31, 32, 33, and 34 elements of F. The third lines contains the 41, 42, 43, and 44 elements of F.

The matrix data should be written in a nested loop where  $\langle h^2 \rangle$  varies more slowly than  $\sigma_z^2$ , which varies more slowly than  $\lambda_0$ , which varies more slowly than  $\theta_0$ . The looping of each variable goes from the first value, to the last value listed in the header (lines C to F). Given this information, DISPLAY can access any of the elements of F on file.

#### AV.7.3.2 Option 12.

Option 12 displays pertinent information associated with the assignment numbers: the material name, auto-correlation function,  $\langle h^2 \rangle$ ,  $\sigma_z^2$ , range on  $\lambda_0$  and range on  $\theta_0$ . This option is simply a reminder option of your choice of the assignment numbers.

#### AV.7.3.3 Option 13.

Option 13 is used to remove an assigned surface and free the assignment number. When the list of assigned surfaces is presented you are prompted to:

Enter the number of the surface you want to remove,  
Enter 0 to return to the main menu,  
Enter -99 to clear out all assignments.

For example, to remove the surface with assignment number 6, enter "6" <RETURN>. To exit without any further removals, enter "0" <RETURN>. Entering "-99" <RETURN> will eliminate all assignments. Option 13 can have an effect on options 21 and 31, should a material in the directive list or the base normalization material be deleted.

#### AV.7.4 PLOT DIRECTIVE LIST MODIFICATIONS.

These routines build lists of information used by the plotting options. Information on plotting options is stored so as to avoid entering repeated data for different plot options.

## AV.7.4.1 Option 21.

Option 21 maintains the list of assignment numbers of the surfaces used in the plotting routines. The first element of this list is the 3-d plot feature. Up to four elements in the assignments list can produce a page of contour plots. The entire list, a maximum of seven, can be used when comparing 2-d contour plots.

The flow of Option 21 is similar to others in this group. On entry, the current list and a menu is displayed. If the list needs to be changed, just enter the number of assignment numbers for a new list. Prompts will guide you to enter a new list of assignment numbers separated by spaces on one line. These new values are checked and, if an illegal input is entered, returned to the start of this option (to reenter the correct list). A return is made to the main menu when the correct inputs are obtained.

## AV.7.4.2 Option 22.

Option 22 controls the list of matrix elements used by the 2-d plotting option. Up to four matrix elements can be drawn on the same page in 2-d format (these element numbers are stored in this list). For example, to plot the matrix elements  $f_{11}$ ,  $f_{22}$ , and  $f_{34}$  on one page, the list will contain "11" and "22" and "34." The method of entering a new list is the same as for Option 21.

## AV.7.4.3 Option 23.

Option 23 maintains a list of wavelengths. These are the fixed wavelengths used when the  $f_{ij}$ 's are plotted versus  $\theta_0$  in the 2-d format. When a comparison is being made between surfaces, only the first wavelength is used, otherwise a comparison can be made between the wavelengths as a function of incident angle. The entry method is the same as above.

## AV.7.4.4 Option 24

Option 24 is like 23, except here a list of fixed incident angles is maintained. This list is used when  $f_{ij}$  is plotted versus  $\lambda_0$  in the 2-d format. The entry method is the same as above.

## AV.7.5 OPTION 31, DATA ANALYSIS OPTIONS.

Two analysis techniques are available in DISPLAY. The first causes all  $f_{ij}$ 's to be divided by the dc element  $f_{11}$ . The second allows one to examine the percent difference between a target surface and a base surface,  $\frac{f_{ij}^{targ}}{f_{ij}^{base}} - 1$ . When both target and base sample measurements are included the normalizations work in succession through division by  $f_{11}$ :

$$\frac{f_{ij}}{f_{11}}, \frac{f_{ij}^{targ}}{f_{ij}^{base}} - 1, \text{ and } \frac{f_{ij}^{targ}}{f_{11}^{targ}} \frac{f_{11}^{base}}{f_{ij}^{base}} - 1.$$

To set up a desired analysis, follow the instructions given in the program.

#### AV.7.5.1 Division by $f_{11}$ .

This feature is included to make the theoretical data more closely resemble normalized or non-normalized Mueller matrix representations (see Section 4.5.4). Division by  $f_{11}$  is performed as the Mueller matrices are being read in as a function of  $\lambda_0$  and  $\theta_0$ , before the comparison to a base material is done.

#### AV.7.5.2 Subtraction and Division of $f_{ij}^{base}$ .

The percent difference simply determines when base and target surfaces, as a function of  $\lambda_0$  and  $\theta_0$ , are most dissimilar. This operation is performed after all the  $f_{ij}$ 's have been read in for the base and the target materials.

#### AV.7.6 OPTION 32, PLOT FORMAT OPTIONS.

This option allows change of various plot features including: page size, line color, tick mark density, grid density, curve thickness, and hidden line removal. The six sub-options are accessed through one menu.

The default page size is 12 by 10.5 inches. However, these dimensions have little to do with the absolute size of the plot coming off the printer. This option lets you change the relative size of the page.

For terminals that have color capabilities supported by Disspla, this program allows one to draw curves on the same set of axes in different colors. This makes comparisons easier, essential for contour plots. By default, multiple lines on the same set of axes are differentiated by line type (dots, dashes, etc.).

The number of tickmarks drawn per major division on the x and y axes can be controlled independently (for aesthetic purposes).

Dashed grid lines can be added to plots to make accurate readings easier. The number of grid lines per major division on both axes can be controlled. Too many grid lines will clutter the plot.

The thickness of curves on 2-d and contour plots can be adjusted depending on the output desired. For terminal display or paper plotting, a thickness of one is usually sufficient. For transparencies, however, the thickness should be increased, especially if color is also used.

The final format option concerns hidden lines on 3-d plots. The default setting does not remove hidden lines (they can clutter the plot). This option must be executed to remove hidden line data in the 3-dimensional plot, or add them in latter plots.

#### AV.7.7 OPTION 41, 3-D PLOTTING.

When executed, this option draws a 3-d representation of  $f_{ij}(\lambda_0, \theta_0)$ . (The 3-d plots are always monochromatic.)

The assignment number of the surface used is obtained from the directive list built by Option 21. The program asks which matrix element is to be plotted (this value is not taken

from the directive list), and the correct reply is row-column element numbers.

The program then requests a range of  $\lambda_0$ 's and  $\theta_0$ 's for the base plane. When a legal base plane range is obtained data that fall within that range are mapped.

The last input required is the viewpoint of the plot. Disspla assumes the plot is in a cube with corners at (0,0,0) and (2,1,1) (arbitrary units). At (0,0,0),  $\lambda_0 = \lambda_0(\min)$  and  $\theta_0 = \theta_0(\min)$ . At (2,1,1),  $\lambda_0 = \lambda_0(\max)$  and  $\theta_0 = \theta_0(\max)$ . The bottom of the cube is  $f_{ij} = f_{ij}(\min)$ . The cube top is  $f_{ij} = f_{ij}(\max)$ . The program requires a viewpoint in terms of these arbitrary box units. Remember, the 1st coordinate is positioned on the  $\lambda_0$  axis, the 2nd coordinate is positioned on the  $\theta_0$  axis, and the 3rd coordinate is positioned on the  $f_{ij}$  axis. The plot and various headings are then drawn. Hidden lines may or may not be removed, depending upon the setting in Option 32.

At this point you have three options: you can change the viewpoint for this matrix element, plot a different matrix element for the same surface and viewpoint, or return to the main menu.

#### AV.7.8 OPTION 42, 2-D PLOTTING.

This option is included because of the difficulty in reading values from 3-d plots, and because of the difficulty in making absolute comparison within and between plots. In general, 2-d plotting allows you to look at cross sections of 3-d plots where either  $\lambda_0$  or  $\theta_0$  has been held constant. The matrix elements plotted are defined by the directive list of Option 22. Up to four matrix elements can be plotted on one page. The four possible plot types, as presented by DISPLAY, are:

1. Spectral with multiple fixed incident angles for 1 surface,
2. Spectral with fixed incident angle for 1+ surfaces,
3. Angular with multiple fixed wavelengths for 1 surface,
4. Angular with fixed wavelength for 1+ surfaces.

Plot type 1 will present  $f_{ij}(\lambda_0, \theta_0^{fixed})$  for up to seven  $\theta_0^{fixed}$  values on the same set of axes. The allowed  $\theta_0^{fixed}$  values are defined in the directive list (Option 24). The range of  $\lambda_0$ 's is entered after plot type 1 or 2 is selected. The first surface assignment number in the surfaces directive list is used.

Plot type 2 presents  $f_{ij}(\lambda_0, \theta_0^{fixed})$  for up to seven different surfaces, defined in the directive list, on the same set of axes.  $\theta_0^{fixed}$  is the first value from Option 24's directive list.

Plot type 3 presents  $f_{ij}(\lambda_0^{fixed}, \theta_0)$  for up to seven  $\lambda_0^{fixed}$  values on the axes. The allowed  $\lambda_0^{fixed}$  values are defined in the directive list in Option 23. The range of  $\theta_0$ 's is entered after plot type 3 or 4 is selected. The first surface assignment number in the surfaces directive list is used.

Plot type 4 draws  $f_{ij}(\lambda_0^{fixed}, \theta_0)$  for up to seven different surfaces, defined in the directive list, on the same set of axes.  $\lambda_0^{fixed}$  is the first value from Option 23's directive list.

Headings and legends are provided on the page to distinguish the curves when comparisons are made. Several plot features can be changed by using Option 31 before starting the plot. The process is the same as for 3-d plots. Press <RETURN> to continue.

Follow the program's instructions before leaving Option 42 to adjust the limits on the

horizontal axis.

*AV.7.9 OPTION 43, CONTOUR PLOTTING.*

Contour plotting is included as a way to read more exact information and to compare materials, without fixing  $\lambda_0$  or  $\theta_0$ . This option draws nine level curves. Color is recommended for this option.

Like the 3-d option, the matrix element and range of the base grid is input at the start of this option. However, unlike the 3-d plots, up to four different surfaces defined in Option 21's directive list can be used. When multiple surfaces are used the level curves are the same for each plot. This means that not all plots will have 9 level curves.

Appropriate headings and legends are drawn to aid your interpretation of the graphs. Several plot features associated with 2-d plots are used, as defined by Option 32.

## Appendix V

### AV.8 DISPLAY SOURCE CODE.

```

1  C This program will plot 2-d, 3-d, and contoured representations of the
2  C Mueller Matrix elements as a function of wavelength and
3  C Incident angle for theoretically calculated values and for
4  C experimental values when the proper input file form is used.
5  C Various surface materials and roughness parameters can be
6  C compared.
7  C Various normalizations will be available for visual analysis.
8  C
9      Integer maxplt,maxcur,maxang,maxwin,maxsur,maxdat
10     Parameter (maxplt=4,maxcur=7,maxang=25)
11     Parameter (maxwin=71,maxsur=10,maxdat=20)
12  C
13     Integer i,j,k,m,l1,l2
14     Integer nhseq,nelga,nwlen(maxsur),nang(maxsur),nkip(maxsur)
15     Integer numplt,numcur,ptype
16     Integer frum(maxplt),surf(maxcur),numpts(maxcur)
17     Integer nangf,nwlenf
18     Integer acode(maxsur)
19     Integer ltemp(maxcur)
20     Integer elt,lcb
21     Integer index(maxplt),ixmr(maxplt),ixmx(maxplt)
22     Integer iymx(maxplt),iymx(maxplt)
23  C
24     Real hneq(maxsur),hmq(15),sigz(maxsur),sigl(15)
25     Real wlen(maxsur,maxwin),angl(maxsur,maxang)
26     Real angfx(maxcur),wlenfx(maxcur)
27     Real xray(maxcur,maxwin),yray(maxcur,maxwin)
28     Real ymr,ymax,xmin,xmax
29     Real xray1(maxwin),yray1(maxwin)
30     Real fct(maxdat,maxwin,maxang)
31     Real fwrk(maxwin,maxang)
32     Real rtemp(maxcur)
33     Real tv(11),dets
34  C
35     Character*13 autorm(8),fname(maxsur),fn
36     Character*13 mainrm(maxsur)
37  C
38  C
39     Common/vone/fct
40  C
41     Data fname/maxsur="none"/
42     Data autorm/'Gaussian','N-8','N-6','Causs. Q only','N-8, Q only',
43     *      'N-6, Q only','P/Q','Experimental'/
44  C
45  C
46  C Initialize main program variables
47     nsurf=0
48     nfrum=0
49     nangf=0
50     nwlenf=0
51  C
52  C Get the terminal being used

```

## Appendix V

```

53      30 write(6,1)
54      write(6," Enter the terminal you are using'
55      write(6," 1) Tek4115'
56      write(6," 2) Tek4107'
57      write(6," 3) HDS200'
58      write(6,")
59      read(5,")term
60      if(term.lt.1.or.term.gt.3) goto 30
61      C
62      C Initialise main program variables and send them off to some subroutines
63      call stpl2(term)
64      call flnk
65      nckx=1
66      ncky=1
67      call stpl3(nckx,ncky)
68      ifcol=0
69      call setlin(ifcol)
70      ifgrd=0
71      ngrdx=0
72      ngrdy=0
73      call agrd2d(ifgrd,ngrdx,ngrdy)
74      call agrdcn(ifgrd,ngrdx,ngrdy)
75      xsize=12.
76      ysize=10.5
77      xscale=1.
78      yscale=1.
79      call sec2d(xscale,yscale)
80      call sec3d(xscale,yscale)
81      call stpl4(xscale,yscale)
82      ndraw=1
83      call stpl5(ndraw)
84      n1=0
85      n2=0
86      call fnorm1(n1)
87      call fnorm2(n2,' ,0,0)
88      ifhide=0
89      call stpl6(ifhide)
90      C
91      C
92      C----- MAIN -----
93      C
94      C Print the main menu and process the command
95      C
96      30 write(6,1)
97      1 format(/)
98      write(6," PLOTTER MAIN MENU'
99      write(6,")
100     write(6," 11. Assign a surface for data access'
101     write(6," 12. Display info. on an assigned surfaces'
102     write(6," 13. Remove an assigned surface'
103     write(6," 21. Modify list of surfaces to use'
104     write(6," 22. Modify list of matrix elements to use'
105     write(6," 23. Modify list of wavelengths to use'
106     write(6," 24. Modify list of incident angles to use'

```

## Appendix V

```

107      write(6," 31. Change analysis normalizations'
108      write(6," 32. Change plotting format features'
109      write(6," 41. Do 3-d plot'
110      write(6," 42. Do 2-d plot'
111      write(6," 43. Do Contour plot'
112      write(6," 99. Quit program'
113      write(6,")
114      write(6," Enter the number of your choice'
115      read(5,"i)
116      if(i.eq.11) goto 100
117      if(i.eq.12) goto 200
118      if(i.eq.13) goto 300
119      if(i.eq.21) goto 400
120      if(i.eq.22) goto 500
121      if(i.eq.23) goto 600
122      if(i.eq.24) goto 700
123      if(i.eq.31) goto 800
124      if(i.eq.32) goto 900
125      if(i.eq.41) goto 1100
126      if(i.eq.42) goto 1000
127      if(i.eq.43) goto 1200
128      if(i.eq.99) goto 1400
129      goto 50
130      C
131      C
132      C***** Option 99 *****
133      C
134      1400 write(6,"Leaving program'
135      stop
136      C
137      C***** Option 11 *****
138      C
139      C This section finds an unused assignment number,
140      C reads in the filename and Q info needed to access the data later, and
141      C saves this information in arrays fname(),acode(),hmax(),sig(),nskip(),
142      C wlen(), and ang().
143      C
144      100 continue
145      c get assign #
146      m=1
147      101 if(fname(m).eq.'none') goto 102
148      m=m+1
149      if(m.le.maxsur) goto 101
150      write(6,1)
151      write(6,"No free space to put surface information'
152      write(6,"Remove a surface then try again'
153      call rcnt
154      goto 30
155      C
156      c get filename and Q info, fn,acode(m)
157      102 write(6,1)
158      write(6,"Enter the material data file name'
159      write(6,"Enter 0 to return to the main menu'
160      read(5,"(a)"/n

```

## Appendix V

```

161      if(fn.eq.'0') goto 50
162      110 write(6,1)
163      write(6,"1. Gauss      2. N-8      3. N-6'
164      write(6,"4. Gauss, Q only 5. N-8, Q only 6. N-6, Q only'
165      write(6,"7. Pij/Q'
166      write(6,"8. Experimental'
167      write(6,")
168      write(6,"Enter the autocorrelation code for surface'
169      read(3,"i)
170      if(i.eq.0) goto 102
171      if(i.lt.1.or.i.gt.8) goto 110
172      acode(m)=i
173      C
174      c Read header info material name, allowed hmaq, sigs, ang and wien
175      write(6,"Opening file ',fn
176      open(unit=10,file=fn,status='old',err=199)
177      read(10,"(a)',end=198)matnum(m)
178      read(10,"*,end=198)nhmaq,nsigs,nwlen(m),nang(m)
179      read(10,"*,end=198)(hmaq(i),i=1,nhmaq)
180      read(10,"*,end=198)(sigs(i),i=1,nsigs)
181      read(10,"*,end=198)(wlen(m,i),i=1,nwlen(m))
182      read(10,"*,end=198)(ang(m,i),i=1,nang(m))
183      C
184      c Get users desired hmaq and sigs
185      c A general data file may have more than one of each
186      120 write(6,1)
187      write(6,"Allowed mean squared heights'
188      write(6,10010)(i,hmaq(i),i=1,nhmaq)
189      10010 format(1x,2(3,' 'e10.3,' '))
190      write(6, )
191      write(6,"Enter the number of your choice'
192      read(3,"j)
193      if(j.lt.1.or.j.gt.nhmaq) goto 120
194      hmaq(m)=hmaq(j)
195      C
196      130 write(6,1)
197      write(6,"Allowed mean squared slopes'
198      write(6,10010)(i,sigs(i),i=1,nsigs)
199      write(6,")
200      write(6,"Enter the number of your choice'
201      read(3,"k)
202      if(k.lt.1.or.k.gt.nsigs) goto 130
203      sigs(m)=sigs(k)
204      C
205      c compute the number of lines to skip past header to get into the correct data
206      c for files generated by RETRO
207      nskip(m)=2*((j-1)*nsigs+k-1)*nwlen(m)*nang(m)
208      C
209      c Print out the results of the assignment operation
210      write(6,"Assigned surface #',m
211      write(6,"To from file',fn
212      write(6,"This mean squared height of',hmaq(m)
213      write(6,"This mean squared slope of',sigs(m)
214      fname(m)=fn

```

## Appendix V

```

215      close(10)
216      call rcort
217      goto 50
218      C
219      c Error - most likely fn spelled wrong
220      199 write(6,"An error occurred opening the file',fn
221      call rcort
222      goto 50
223      C
224      c Error - most likely RETRO didn't finish putting in data
225      198 write(6,"Not enough data on unit 10 for file',fn
226      call rcort
227      goto 50
228      C
229      C***** Option 12 *****
230      C
231      C This section prints out all the current assignments
232      C The specified autocorrelation,hmaq,sigs and stored wlen and ang ranges
233      C are displayed to refresh to users memory about assignments
234      200 continue
235      write(6,1)
236      write(6," SURFACE ASSIGNMENTS'
237      write(6,"
238      do 220 i=1,maxsur
239      if(trim(name(i)).eq.'none')then
240      write(6,10210)
241      10210 format(13,' - surface not assigned')
242      else
243      write(6,10220),matnum(i),autonm(acode(i))
244      write(6,10230)hmaq(i),sigs(i)
245      write(6,10240)wlen(i,1),wlen(i,nwlen(i)),
246      *      ang(i,1),ang(i,nang(i))
247      10220 format(13,' - material - ',a20,' corr. fac. - ',a)
248      10230 format(7x,'mean sq. height - ',f8.3,' mean sq. slope - ',f8.3)
249      10240 format(7x,'wlen min,max ',2f8.3,' inc ang min,max',2f7.2)
250      endif
251      write(6,"
252      c put a page break every 5 assignment numbers
253      if(i.eq.5.or.i.eq.10.or.i.eq.15) call rcort
254      220 continue
255      goto 50
256      C
257      C***** Option 13 *****
258      C
259      C This routine frees assignment numbers for use with other surfaces
260      C This operation can affect options 21 and 31, and subroutine freads.
261      C The current assignments are displayed and the menu given.
262      C The user's option is read and executed
263      300 continue
264      write(6,1)
265      write(6," ASSIGNED SURFACES'
266      write(6,"
267      write(6,10319)'Assign # ',Material ',Corr. fac. ',
268      *      ' mean sq hgt',mean sq slope'

```

## Appendix V

```

269 10319 format(1x,a8,2x,a13,2x,a13,2x,a13,2x,a13)
270 do 303 i=1,maxsur
271   if(name(i).ne.'none') then
272     write(6,10320),matnum(i),autonm(acode(i)),hmag(i),sig(i)
273   endif
274 10320 format(1x,i8,2x,a13,2x,a13,2x,f13.3,2x,f13.3)
275 303 continue
276   write(6,*)
277 320 continue
278   write(6,*)'Enter the number of the surface you want to remove'
279   write(6,*)'Enter 0 to return to the main menu'
280   write(6,*)'Enter -99 to clear out all assignments'
281   read(5,*) i
282   if(i.eq.-99) goto 390
283   if(i.eq.0) goto 50
284   if(i.lt.1.or.i.gt.maxsur) goto 300
285 C
286 c Free assignment number i only and update freadz
287 c Check to see if options 31 or 21 are affected
288   frame(i)='none'
289   call fupdt(i)
290   if(i.eq.n2)then
291     write(6,*)'Base surface for normalizations has been deleted'
292     write(6,*)'Use option 31 to reassign base surface'
293     write(6,*)
294     n2=0
295   endif
296   do 380 j=1,nsurf
297     if(surf(j).eq.i) then
298       nsurf=nsurf-1
299       write(6,*)'The surfaces to use list has been cleared.'
300       write(6,*)'Use main menu option 21 to rebuild'
301       call rcost
302     endif
303 380 continue
304   goto 300
305 C
306 c Free all assignment numbers and initialize freadz
307 c Option 21 is affected and 31 might be too
308 390 do 392 i=1,maxsur
309   frame(i)='none'
310 392 continue
311   call fndt
312   if(n2.ne.0) then
313     write(6,*)'Base surface for normalisations has been deleted'
314     n2=0
315   endif
316   nsurf=0
317   write(6,*)'The surfaces to use list has been cleared.'
318   write(6,*)'Use main menu option 21 to rebuild'
319   call rcost
320   goto 50
321 C
322 C===== Option 21 =====

```

## Appendix V

```

323      C
324      C The list of surfaces to be used in some 2-d and contour plots is built here
325      C The 1st assignment # in the list is used by some 2-d and the 3-d routine
326      C The current surfaces 'to use' list is printed
327      C The menu is given, the user's choice read, and executed
328      400 continue
329      write(6,1)
330      write(6,*)      LIST OF SURFACES TO USE'
331      write(6,*)
332      write(6,10419)'List # ','Assign #','Material ','
333      *      'Corr. fnc. ','mean sq hgt','mean sq slope'
334      10419 format(a7,2x,a8,2x,a13,2x,a13,2x,a13,2x,a13)
335      do 403 i=1,nsurf
336      m=surf(i)
337      write(6,10420)i,m,matnm(m),autnm(aicode(m)),hmq(m),sigs(m)
338      10420 format(4,'-',i8,2x,a13,2x,a13,2x,f13.3,2x,f13.3)
339      403 continue
340      write(6,*)
341      411 write(6,*)
342      write(6,*)'Enter # of surfaces for list or'
343      write(6,*)'enter 0 to return to the main menu'
344      read(3,*)j
345      if(j.eq.0) goto 50
346      if(j.lt.1.or.j.gt.maxsur) goto 411
347      C
348      c read the number of assignment numbers requested into a temp array
349      420 write(6,*)
350      write(6,*)'Enter the numbers of the assigned surfaces'
351      write(6,*)'you want in the list. Enter in desired order'
352      write(6,*)'separated by spaces on one line.'
353      write(6,*)'Enter all zeros to get out.'
354      read(3,*)(itemp(i),i=1,j)
355      do 430 i=1,j
356      if(itemp(i).eq.0) goto 400
357      if(itemp(i).lt.1.or.itemp(i).gt.maxsur) goto 420
358      if(iname(itemp(i)).eq.'none') then
359      write(6,*)'At least one surface is not assigned'
360      goto 420
361      endif
362      430 continue
363      C
364      c if all values are legitimate copy them into storage array surf()
365      nsurf=j
366      do 440 i=1,nsurf
367      440 surf(i)=itemp(i)
368      goto 400
369      C
370      C----- Option 22 -----
371      C
372      C This option builds the matrix elements 'to use' list that is necessary
373      C to do 2-d plots with multiple mix elements
374      C The current list is presented in mix notation P11, P12, P44 etc.,
375      C however the data is stored linearly P11-P(1),P12-P(2),P44-P(16)
376      C The menu is given, the choice read, and executed.

```

## Appendix V

```

377      500 continue
378      write(6,1)
379      write(6,"      LIST OF MATRIX ELEMENTS TO USE"
380      write(6,"
381      do 510 i=1, nfrum
382          i1=(frum(i)-1)/4+1
383          i2=frum(i)-(i1-1)*4
384          write(6,10510)i,i1,i2
385 10510 format(2x,i8,' - P',2i1)
386      510 continue
387      511 write(6,"
388          write(6,"Enter number of matrix elements for list or'
389          write(6,"enter 0 to return to the main menu'
390      read(5," )
391      if(j.eq.0) goto 50
392      if(j.lt.1.or.j.gt.maxplt) goto 511
393      C
394      c This reads the mtr notation for the elt's and converts to linear format
395      515 write(6,"
396          write(6,"Enter the matrix element codes for the list'
397          write(6,"Enter them in desired order separated by spaces'
398          write(6,"e.g. if you want P11,P24, and P34 then enter'
399          write(6,"11 22 34'
400      write(6,"
401      read(5,")(ltemp(i),i=1,j)
402      do 520 i=1,j
403          if(ltemp(i).eq.0) goto 500
404          ltemp(i)=ltemp(i)-6*(ltemp(i)/10)-4
405          if(ltemp(i).lt.1.or.ltemp(i).gt.16) goto 515
406      520 continue
407      C
408      c Copy the temp array into to the storage array
409      nfrum=j
410      do 540 i=1,nfrum
411          540 frum(i)=ltemp(i)
412      goto 500
413      C
414      C***** Option 23 *****
415      C
416      C This section reads the list of fixed wavelengths to be used by
417      C the 2-d routine when cross sections of 3-d plots are made
418      C parallel to the inc. angle axis.
419      C The current list is given in micrometers, the menu is plotted,
420      C the choice read and executed
421      600 continue
422      write(6,1)
423      write(6,"      WAVELENGTHS TO USE"
424      write(6,"
425      do 610 i=1,nwlenf
426          write(6,10600)i,wlenfx(i)
427 10600 format(2x,i4,' - ',f7.3)
428      610 continue
429      611 write(6,"
430          write(6,"Enter the number of wavelengths for list or'

```

## Appendix V

```

431      write(6,*)'enter 0 to return to the main menu'
432      read(5,*)j
433      if(j.eq.0) goto 50
434      if(j.lt.1.or.j.gt.maxcur) goto 611
435      C
436      c The desired number of wavelengths are read into a temp array
437      615 write(6,*)
438      write(6,*)'Enter the desired wavelengths in micro-meters'
439      write(6,*)'separated by spaces on one line'
440      read(5,*)(rtemp(i),i=1,i)
441      do 620 i=1,i
442      if(rtemp(i).eq.0.) goto 600
443      if(rtemp(i).lt.0.) goto 615
444      620 continue
445      C
446      c The temp array is copied if all values that are legitimate.
447      nwlent=j
448      do 640 i=1,nwlent
449      640 wlenfx(i)=rtemp(i)
450      goto 600
451      C
452      C===== Option 24 =====
453      C
454      C This section reads in the list of inc angles to be used by the 2-d routine
455      C when cross sections parallel to the wavelength axis are being made.
456      C The current list is presented with the menu.
457      C The choice is read and executed.
458      700 continue
459      write(6,1)
460      write(6,*)      INCIDENT ANGLES TO USE'
461      write(6,*)
462      do 710 i= 1,nangf
463      write(6,10700),angfx(i)
464      10700 format(2x,i4,' - ',f7.2)
465      710 continue
466      711 write(6,*)
467      write(6,*)'Enter the number of incident angles for list or'
468      write(6,*)'enter 0 to return to main menu'
469      read(5,*)j
470      if(j.eq.0) goto 50
471      if(j.lt.1.or.j.gt.maxcur) goto 711
472      C
473      c Read values into a temp array and check them
474      715 write(6,*)
475      write(6,*)'Enter the desired incident angles in degrees'
476      write(6,*)'separated by spaces on one line'
477      read(5,*)(rtemp(i),i=1,i)
478      do 720 i=1,i
479      if(rtemp(i).lt.0.) goto 700
480      if(rtemp(i).gt.90.) goto 715
481      720 continue
482      C
483      c Copy temp array into storage if all vals OK
484      nangf=j

```

## Appendix V

```

485      do 740 i=1,nangf
486      740 angfx(i)=rtemp(i)
487      goto 700
488      C
489      C***** Option 31 *****
490      C
491      C Write out current set up for normalizations
492      C Display menu
493      C Get user command
494      800 continue
495      write(6,1)
496      write(6,*)  ANALYSIS NORMALIZATIONS'
497      write(6,*)
498      write(6,*)  Current set up:'
499      if(n1.eq.1) then
500      write(6,*)  Pij -> Pij/F11'
501      write(6,*)  All F11 elements scaled -> 1'
502      write(6,*)
503      endif
504      if(n2.ne.0) then
505      write(6,*)  Pij(object) -> Pij(object)/Pij(base)-1'
506      write(6,*)  where the base surface is '
507      write(6,10805)'Assign','Material','Auto corr','mean sq hgt',
508      *      'mean sq slope'
509      10805 format(a8,a15,a15,a15,a15)
510      write(6,10810)n2,matnm(n2),autonm(acode(n2)),hmsq(n2),sigs(n2)
511      10810 format(i8,a15,a15,f15.3,f15.3)
512      write(6,*)
513      else if(n1.eq.0) then
514      write(6,*)  No normalizations in place'
515      write(6,*)
516      endif
517      C
518      810 write(6,*)Enter 1 for Pij -> Pij/F11'
519      write(6,*)Enter 2 to cancel division by F11 element'
520      write(6,*)Enter 3 for Pij(object) -> Pij(object)/Pij(base)-1'
521      write(6,*)Enter 4 to cancel comparisons with a base surface'
522      write(6,*)Enter 0 to return to the main menu'
523      write(6,*)
524      read(5,*)i
525      if(i.eq.0) goto 50
526      if(i.lt.1.or.i.gt.4) goto 810
527      C
528      c n1=1 -> divide Pij by F11 at readin in readsf
529      c and pass this along through fnorm1
530      if(i.eq.1) then
531      n1=1
532      call fnorm1(n1)
533      C
534      c n1=0 -> do NOT divide by F11
535      else if(i.eq.2) then
536      n1=0
537      call fnorm1(n1)
538      C

```

## Appendix V

```

539   c n2=0 —> there is no base material
540   c and reset this for fnorm2
541   else if(l.eq.4) then
542     n2=0
543     call fnorm2(n2,' ',0,0)
544   C
545   c n2>0 —> there is a base material and its assignment number is n2
546   c print menu and get base material's assignment number if desired.
547   c pass information to fnorm2
548   else if(l.eq.3) then
549     820   write(6,*)
550           write(6,*)"Enter the number of the assignment # of the surface"
551           write(6,*)"you want to use as the base or"
552           write(6,*)"Enter 0 to back up a menu"
553           read(5,*)j
554           if(j.eq.0) goto 810
555           if(j.lt.0.or.j.gt.maxsur) goto 820
556           if(fname(j).eq.'none') then
557             write(6,*)"That surface is not assigned"
558             goto 820
559           endif
560           n2=j
561           call fnorm2(n2,fname(j),acode(j),nskip(j))
562         endif
563   c changing the normalization makes all old data obsolete (use finit)
564   call finit
565   goto 800
566   C
567   C***** Option 32 *****
568   C
569   C This section allows modification of page size, color on/off,
570   C tickmark density, grid lines, curve thickness, and hidden line removal.
571   C The current setup is displayed in the menu.
572   900 continue
573   write(6,1)
574   write(6,*)   CURRENT PLOT PARAMETERS
575   write(6,*)
576   write(6,10910)pxsize,ysize
577   10910 format(' The plot page is ',f4.1,'in. by ',f4.1,'in.')
578   write(6,*)
579   c
580   if(ifcol.eq.1) then
581     write(6,*)"Curves distinguished by color"
582   else
583     write(6,*)"Curves distinguished by line type"
584   endif
585   write(6,*)
586   c
587   write(6,10915)ntickx,nticky
588   10915 format(' There are ',i1,' tick marks per major x-axis division',
589   *   /,' There are ',i1,' tick marks per major y-axis division')
590   write(6,*)
591   c
592   if(ifgrd.eq.1) then

```

## Appendix V

```

593      write(6,*)'Grid option for 2-d and contour plot is in effect'
594      write(6,10920)ngrdx,ngrdy
595 10920 format(' There are ',i1,' grid lines per major tick on the',
596      * ' x-axis',/, ' There are ',i1,' grid lines per major tick',
597      * ' on the y-axis')
598      else
599      write(6,*)'No grid lines'
600      endif
601      write(6,*)
602  c
603      write(6,10930)ndraw
604 10930 format(' Curves on 2-d and contour plots are drawn',i2,' times')
605      write(6,*)
606  c
607      if(lhide.eq.0) then
608      write(6,*) 'On 3-d plots all lines drawn'
609      else
610      write(6,*) 'On 3-d plots hidden lines removed'
611      endif
612  C
613      write(6,*)
614      write(6,*)
615  920 write(6,*)'Enter 1 to change page size'
616      write(6,*)'Enter 2 to change color/line-type option'
617      write(6,*)'Enter 3 to change the tick mark density'
618      write(6,*)'Enter 4 to modify the grid option'
619      write(6,*)'Enter 5 to change curve thickness on 2-d and//'
620      * 'Contour plots'
621      write(6,*)'Enter 6 to modify 3-d hidden lines option'
622      write(6,*)'Enter 0 to return to the main menu'
623      write(6,*)
624      read(3,*)i
625      if(i.eq.0) goto 50
626      if(i.lt.1.or.i.gt.6) goto 920
627  C
628      write(6,*)
629  c This section reads in new page size xsize,ysize
630  c then resets the scale and passes then info via stp14, sec2d, and sec3d
631      if(l.eq.1) then
632      930 write(6,*)'The normal page size is 12in by 10.5in'
633      write(6,*)'Enter the x and y size of the page in inches'
634      write(6,*)
635      read(3,*)xsize,ysize
636      if(xsize.lt.0.0.or.ysize.lt.0.0) goto 930
637      xscale=xsize/12.
638      yscale=ysize/10.5
639      call stp14(xscale,yscale)
640      call sec2d(xscale,yscale)
641      call sec3d(xscale,yscale)
642  C
643  c This option changes the way lines are distinguished
644  c ifcol=1 --> use color option, otherwise use line type option
645  c Write menus get a valid response for ifcol and pass along to setlin
646      else if(l.eq.2) then

```

## Appendix V

```

647         if(Iterm.eq.3) then
648             write(6,*)'Color option not available on HDS200'
649         else
650     935     write(6,*)'Enter 1 to distinguish curves by color'
651             write(6,*)'Enter 2 to distinguish curves by line-type'
652             read(5,*)i
653             if(i.lt.1.or.i.gt.2) goto 935
654             if(i.eq.1) then
655                 ifcol=1
656             else
657                 ifcol=0
658             endif
659             call setlin(ifcol)
660         endif
661     C
662     c This allows changing the total number of tick marks between in a major divis
663     c Get legal values for x and y axes then pass along to stplt3
664     else if(l.eq.3) then
665     940     write(6,*)'Enter the number of tick marks per major division'
666             write(6,*)'along the x and y axes in the allowed range 1..9'
667             write(6,*)
668             read(5,*)nxcix,nicky
669             if(nxcix.lt.1.or.nxcix.gt.9) goto 940
670             if(nicky.lt.1.or.nicky.gt.9) goto 940
671             call stplt3(nxcix,nicky)
672     C
673     c This allows removing grid lines all together or controlling the number to draw
674     c get legal values the ifgrd, ngrdx, and ngrdy
675     c ifgrd=0 --> no grids otherwise use ngrdx and ngrdy as the number to use
676     c pass this info to sgrd2d and sgrdcn
677     else if(l.eq.4) then
678     945     write(6,*)'Enter 1 to turn the grid option off'
679             write(6,*)'Enter 2 to turn on grid option and modify density'
680             write(6,*)
681             read(5,*)i
682             if(i.lt.1.or.i.gt.2) goto 945
683             if(i.eq.1) then
684                 ifgrd=0
685             else
686                 ifgrd=1
687     947     write(6,*)'Enter the number of grid lines per major division'
688             write(6,*)'along the x and y axes in the allowed range 0..9'
689             write(6,*)
690             read(5,*)ngrdx,ngrdy
691             if(ngrdx.lt.0.or.ngrdx.gt.9) goto 947
692             if(ngrdy.lt.0.or.ngrdy.gt.9) goto 947
693             endif
694             call sgrd2d(ifgrd,ngrdx,ngrdy)
695             call sgrdcn(ifgrd,ngrdx,ngrdy)
696     C
697     c This allows the user to change the number of times the curve is
698     c drawn over the same path. This allows thicker curves especially for
699     c making transparencies. Get info in ndraw and pass to stplt5.
700     else if(l.eq.5) then

```

## Appendix V

```

701      950  write(6,*)'Enter the number of times the curves should be'
702          write(6,*)'drawn to control thickness'
703          write(6,*)'The allowed range is 1..5'
704          read(3,*)ndraw
705          if(ndraw.lt.1.or.ndraw.gt.5) goto 950
706          call stph5(ndraw)
707  C
708  c This allow the user to remove hidden lines from 3-d plots or to put them
709  c back in.  ihide-1 --> remove hidden lines. Pass info to stph6.
710          else if(i.leq.6) then
711      960  write(6,*)'Enter 1 to remove hidden lines from 3-d plots'
712          write(6,*)'Enter 2 to draw all lines on 3-d plots'
713          read(3,*)i
714          if(i.lt.1.or.i.gt.2) goto 960
715          if(i.eq.1) then
716              ihide-1
717          else
718              ihide-0
719          endif
720          call stph6(ihide)
721          endif
722          goto 900
723  C
724  C***** Option 41 *****
725  C
726  C This option plots 2-d representations of data
727  c first make sure there is a material and a matrix element to plot
728      1000 continue
729          if(nsurf.lt.1) then
730              write(6,*)'No surfaces in the "to use" list'
731              write(6,*)'Use main menu option 21 and try again'
732              call rcont
733              goto 50
734          endif
735  C
736          if(nfnum.lt.1) then
737              write(6,*)'No matrix elements in the to use list'
738              write(6,*)'Use main menu option 22 and try again'
739              call rcont
740              goto 50
741          endif
742  C
743  c print menu and get response
744  c Options 1 and 3 use only the 1st element in surfaces list but they use
745  c the entire angle list and wavelength list respectively to determine
746  c how many curves should be drawn per mtr element.
747  c Options 2 and 4 use the surfaces list to determine how many curves should be
748  c drawn per mtr element and the 1st element of the angle list and wavelength
749  c list respectively are used to fix that parameter to compare surfaces.
750      1001 write(6,1)
751          write(6,*)'Enter the type of plot you want'
752          write(6,*)' 1. Spectral with multiple fixed inc. angles '
753          *      // 'for 1 surface'
754          write(6,*)' 2. Spectral with a fixed inc. angle for 1+ surfaces'

```

## Appendix V

```

755      write(6,*) ' 3. Angular with multiple fixed wavelengths '
756      *      // 'for 1 surface'
757      write(6,*) ' 4. Angular with a fixed wavelength for 1+ surfaces'
758      write(6,*)
759      read(5,*) ptype
760      if(ptype.eq.0) goto 50
761      if(ptype.lt.1.or.ptype.gt.4) goto 1001
762      C
763      c Options 1 and 2 make sure there is at least one fixed inc angle
764      iflag3=0
765      if(ptype.lt.3) then
766          if(nangf.lt.1) then
767              write(6,*) 'No fixed incident angles in list'
768              write(6,*) 'Use main menu option 24 and try again'
769              call rcont
770              goto 50
771          endif
772      C
773      c Options 3 and 4 make sure there is at least one fixed wavelength
774      else
775          if(nwlenf.lt.1) then
776              write(6,*) 'No fixed wavelengths in list'
777              write(6,*) 'Use main menu option 23 and try again'
778              call rcont
779              goto 50
780          endif
781      endif
782      C
783      c Get range for the x-axis of the plots
784      1002 write(6,1)
785      if(ptype.lt.3) then
786          write(6,*) 'Enter the range on wavelength in micro-meters'
787          write(6,*) 'Minimum and maximum separated by spaces on one line'
788          read(5,*) xmin,xmax
789          if(xmin.lt.0.0.or.xmin.ge.xmax) goto 1002
790      else
791          write(6,*) 'Enter the range on incident angle in degrees.'
792          write(6,*) 'Minimum and maximum separated by spaces on one line'
793          read(5,*) xmin,xmax
794          if(xmin.lt.0.0.or.xmax.gt.90.0.or.xmin.ge.xmax) goto 1002
795      endif
796      c iflag3=1 -> Plot already done once and only a change in x-range desired
797      if(flag3.eq.1) goto 1012
798      C
799      C
800      c Print out in words what will be plotted to allow the user to back out
801      write(6,1)
802      write(6,*) ' PLOT SETUP'
803      write(6,*)
804      if(ptype.eq.1.or.ptype.eq.3) then
805          m=surf(1)
806          write(6,11062) 'List #', 'Assign #', 'Material', 'Corr. fac.',
807      *      'mean eq hgt', 'mean eq slope'
808          write(6,11063) 1,m,matnm(m),autnm(acode(m)),hmag(m),slga(m)

```

## Appendix V

```

809      write(6,*)
810    endif
811  C
812    write(6,11010)nfrum
813    do 1003 i=1,nfrum
814      l1=(frum(i)-1)*4+1
815      l2=frum(i)-(l1-1)*4
816    1003 write(6,11020)l1,l2
817    11010 format(' There will be ',l3,' mtr. elements displayed')
818    11020 format('110, ' - F',2i1)
819    write(6,*)
820  C
821    if(ptype.gt.2) then
822      write(6,11030)xmin,xmax
823    else
824      write(6,11040)xmin,xmax
825    endif
826    11030 format(' Each mtr. element will be plotted vs. incident angle',
827      *      ' /,' in the range from ',f5.1,' to ',f5.1,' degrees')
828    11040 format(' Each mtr. element will be plotted vs. wavelength',
829      *      ' /,' in the range from ',f7.3,' to ',f7.3,' micro-meters')
830  C
831    write(6,*)
832    if(ptype.eq.1) then
833      write(6,11050)nangf
834      do 1004 i=1,nangf
835    1004 write(6,11052)l,angfx(i)
836    11050 format(' For each mtr. element ',l3,' fixed incident angles',
837      *      ' ' will be used')
838    11052 format('110, ' - ',f5.1)
839  C
840    else if(ptype.eq.3) then
841      write(6,11054)nwlentf
842      do 1006 i=1,nwlentf
843    1006 write(6,11056)l,wlenfx(i)
844    11054 format(' For each mtr. element ',l3,' fixed wavelengths',
845      *      ' ' will be used')
846    11056 format('110, ' - ',f7.3)
847  C
848    else
849      write(6,11058)naurf
850      write(6,*)
851      write(6,11062)'List #','Assign #','Material','Corr. fac.',
852      *      'mean eq hgt','mean eq slope'
853      do 1008 i=1,naurf
854      m=aurf(i)
855      write(6,11063)l,m,matnum(m),autnum(acode(m)),hmasq(m),sigm(m)
856    1008 continue
857      write(6,*)
858      if(ptype.eq.2) then
859        write(6,11060)angfx(1)
860      else
861        write(6,11061)wlenfx(1)
862      endif

```

## Appendix V

```

863         endif
864 11058 format(' For each mtr. element ',i3,' surfaces will be used.')
865 11060 format(' The fixed angle of incidence will be ',f5.1)
866 11061 format(' The fixed wavelength will be ',f7.3)
867 11062 format(a8,a8,2x,a13,2x,a13,2x,a13,2x,a13)
868 11063 format(14,' - ',i8,2x,a13,2x,a13,2x,f13.3,2x,f13.3)
869 C
870 c If the plot set up is correct continue otherwise back up
871     write(6,*)
872 1009 write(6,*)"Enter 1 if this is correct"
873     write(6,*)"Enter 0 to return to the main menu"
874     read(5,*)i
875     if(i.eq.1) goto 1010
876     if(i.eq.0) goto 50
877     goto 1009
878 C
879 C
880 c numplt is the number of mtr elements to be use (up to 4)
881 c numcur is the number of curves per mtr elt dependent upon plot type (ptype)
882 1010 continue
883     numplt=nnum
884     if(ptype.eq.1) then
885         numcur=nangf
886     else if(ptype.eq.3) then
887         numcur=nwlenf
888     else
889         numcur=nsurf
890     endif
891 C
892 c iflag1=-1 --> only one surface material at one roughness scale
893 c iflag1=0 --> comparison of roughness parameters for one material
894 c iflag1=1 --> comparison of materials and roughness is assumed to be same
895 c iflag2=-1 --> only one fixed inc angle or fixed wavelength is used
896 c iflag2=0 --> multiple curves correspond to 1+ fixed wavelengths
897 c iflag2=1 --> multiple curves correspond to 1+ fixed inc angles
898     if(numcur.eq.1) then
899         iflag1=-1
900         iflag2=-1
901     else
902         if(ptype.eq.2.or.ptype.eq.4) then
903             if(matnm(surf(1)).eq.matnm(surf(2)).and.
904             * acode(surf(1)).eq.acode(surf(2))) then
905                 iflag1=1
906             else
907                 iflag1=0
908             endif
909             iflag2=-1
910         else
911             if(ptype.eq.1) then
912                 iflag2=1
913             else
914                 iflag2=0
915             endif
916             iflag1=-1

```

## Appendix V

```

917         endif
918     endif
919 C
920 c This is the section where the actual plotting is called for
921 1012 call stplot
922     m=surf(1)
923 c this call draws heading display in the middle of the page
924     call head2d(numplt,matnm(m),autnm(icode(m)),hmq(m),
925 *         sigs(m),wlenfx(1),angfx(1),iflag1,iflag2,ptype)
926 c this section decides which, if any, legends need to be drawn to
927 c distinguish the curves from one another
928     if(iflag1.eq.0) then
929         call lgmat(numcur,matnm,maxsur,surf,icode,autnm)
930     else if(iflag1.eq.1) then
931         call lgrut(numcur,hmq,sigs,maxsur,surf)
932     else if(iflag2.eq.0) then
933         call lgwin(numcur,wlenfx)
934     else if(iflag2.eq.1) then
935         call lgang(numcur,angfx)
936     endif
937 c If a base normalization is in effect then give its stats in a legend
938     if(n2.ne.0) then
939         call lgbase(matnm(n2),autnm(icode(n2)),hmq(n2),sigs(n2))
940     endif
941 C
942 c This section calls the data management routines to read in data files.
943 c one or more calls may be necessary to freads
944     if(ptype.eq.2.or.ptype.eq.4) then
945         do 1013 i=1,nsurf
946             m=surf(i)
947             call freads(m,nfnum,fnum,fname(m),icode(m),nkip(m))
948 1013 continue
949         else
950             m=surf(1)
951             call freads(m,nfnum,fnum,fname(m),icode(m),nkip(m))
952         endif
953 C
954 c this is the main loop for each mtr element plot draw all curves in succession
955 c carve extracts the curve data from storage, interpolates as necessary, and
956 c puts the values to be plotted in xray and yray.
957 c The first index in xray(..) and yray(..) is the curve number
958     do 1020 i=1,numplt
959         if(ptype.eq.2.or.ptype.eq.4) then
960             do 1025 j=1,numcur
961                 m=surf(j)
962                 call carve(j,m,fnum(i),angl,wlenl,nang(m),nwlen(m),
963 *                     angfx,wlenfx,ptype,numpts,xray,yray)
964 1025 continue
965             else
966                 m=surf(1)
967                 call carve(1,numcur,m,fnum(i),angl,wlenl,nang(m),nwlen(m),
968 *                     angfx,wlenfx,ptype,numpts,xray,yray)
969             endif
970 C

```

## Appendix V

```

971      c This section finds min and max y value and sets up limits for plots
972      c Then the physical origin and scale of plot is set
973      c the axes are drawn and labeled
974      call mg2d(numcur,xray,xmin,xmax,yray,numpts,maxcur,maxwin,
975      *      ymin,ymax)
976      call prin2d(numpt,i)
977      call axes2d(ptype,fnum(i),xmin,xmax,ymin,ymax,n1,n2)
978      do 1030 j=1,numcur
979      c for each curve change the line's appearance is changed and the curve drawn
980      call lines(j)
981      call dot2d(j,numpts(j),xray,yray,xray1,yray1,maxcur,maxwin)
982      1030 continue
983      call engr2d
984      1020 continue
985      call enplot
986      C
987      1040 write(6,*)'Enter 1 to change x-axis range for same plot format'
988      write(6,*)'Enter 0 to return to the main menu'
989      read(3,*)i
990      if(i.eq.1) goto 1002
991      if(i.eq.0) goto 50
992      goto 1040
993      C
994      C***** Option 42 *****
995      C
996      C This section does 3-d plotting
997      c make sure there is a surface to use
998      1100 continue
999      if(nsurf.lt.1) then
1000      write(6,*)'No surfaces in the to use list'
1001      write(6,*)'Use main menu option 21 and try again'
1002      call rcnt
1003      goto 50
1004      endif
1005      iflag1=0
1006      C
1007      c get the desired mtr element
1008      1101 write(6,1)
1009      write(6,*)'Enter the mtr element position you want using'
1010      write(6,*)'the matrix notation'
1011      write(6,*)'e.g. if you want F11, F22, or F34 then enter//'
1012      *      ' 11 or 22 or 34'
1013      write(6,*)'Enter 0 to return to the main menu'
1014      read(3,*)i
1015      if(i.eq.0) goto 50
1016      ek=i-6*(i/10)-4
1017      if(ek.lt.1.or.ek.gt.16) goto 1101
1018      c iflag1=1 -> view pt and base range stays the same but mtr element changes
1019      c skip unnecessary menus
1020      if(iflag1.eq.1) goto 1128
1021      C
1022      1102 iflag2=0
1023      write(6,1)
1024      write(6,11110)

```

## Appendix V

```

1025 11110 format(' This plot will be for mtr. element P',I2)
1026 write(6,")
1027 C
1028 write(6,11062)'List  $\theta$ ','Assign  $\theta$ ','Material','Corr. fnc.',
1029 'mean sq hgt','mean sq slope'
1030 m=surf(1)
1031 write(6,11063)1,m,matrun(m),autonun(icode(m)),hmax(m),sigs(m)
1032 write(6,")
1033 C
1034 c get range for x-axis which has wavelength info
1035 1115 write(6,1)
1036 write(6,")'Enter the range on wavelength (min max)'
1037 write(6,")'Enter 1 1 for auto ranging'
1038 read(3,")xmin,xmax
1039 if(xmin.eq.xmax) goto 1120
1040 if(xmin.lt.0.0.or.xmin.ge.xmax) goto 1115
1041 C
1042 c get range for y-axis which has the inc angle info
1043 1120 write(6,")'Enter the range on incident angle (min max)'
1044 write(6,")'Enter 1 1 for auto ranging'
1045 read(3,")ymin,ymax
1046 if(ymin.eq.ymax) goto 1125
1047 if(ymin.lt.0.0.or.ymax.gt.90.0.or.ymin.ge.ymax) goto 1120
1048 C
1049 c get view point in terms of the plot box size and location
1050 1125 write(6,")'Enter the vu point (xpos,ypos,zpos) in box units'
1051 write(6,")'The display box has corners at (0,0,0) and (2,1,1)'
1052 write(6,")'Box point (0,0,0) has values (x,y,z)-(xmin,ymin,zmin)'
1053 write(6,")'Box point (2,1,1) has values (x,y,z)-(xmax,ymax,zmax)'
1054 write(6,")'x - wavelength, y - inc. ang. z - Pij'
1055 write(6,")'Enter 0 0 0 to back up'
1056 write(6,")'Enter -3 3 3 for normal viewing'
1057 read(3,")vx,vy,vz
1058 if(vx.eq.0.0.and.vy.eq.0.0.and.vz.eq.0.) goto 1102
1059 if((0.le.vx.and.vx.le.2.).and.(0.le.vy.and.vy.le.1.).and.
1060 '(0.le.vz.and.vz.le.1.)) then
1061 write(6,")'Do not put the view point inside the display box'
1062 write(6,")
1063 goto 1125
1064 endif
1065 c flag2-1 --> base range and mtr elt stayed the same but the view pt changed
1066 c skip unnecessary operations
1067 if(flag2.eq.1) goto 1150
1068 C
1069 1109 write(6,")'Enter 1 if this is correct'
1070 write(6,")'Enter 0 to return to the main menu'
1071 read(3,")i
1072 if(i.eq.1) goto 1128
1073 if(i.eq.0) goto 50
1074 goto 1109
1075 C
1076 C
1077 c this makes sure the proper data is available
1078 1128 m=surf(1)

```

## Appendix V

```

1079      j=fnum(1)
1080      fnum(1)=elt
1081      call freads(m,1,fnum,frame(m),acode(m),nskip(m))
1082      fnum(1)=j
1083      call find(m,elt,idx)
1084      write(6,"j m,elt,idx",m,elt,idx)
1085      C
1086      c this determines which data pts are needed to cover the user's range,
1087      c otherwise autoranging takes all data points.
1088      if(xmin.eq.xmax) then
1089          bxmin=1
1090          bxmax=nwlen(m)
1091      C
1092      c otherwise find smallest x range that covers the user's range
1093      else
1094          bxmin=1
1095      1160  if(xmin.gt.wlen(m,bxmin).and.bxmin.lt.nwlen(m)) then
1096          bxmin=bxmin+1
1097          goto 1160
1098      endif
1099      bxmax=nwlen(m)
1100      1162  if(xmax.lt.wlen(m,bxmax).and.bxmax.gt.1) then
1101          bxmax=bxmax-1
1102          goto 1162
1103      endif
1104      endif
1105      if(bxmin.gt.1) bxmin=bxmin-1
1106      if(bxmax.lt.nwlen(m)) bxmax=bxmax+1
1107      c reset range to correspond to plot
1108      c DISPLA should support an unevenly space base grid through the SURTRN
1109      c command but I couldn't get it to work so the xmin and xmax range need
1110      c to be reset so even base grid spacing can be used.
1111      xmin=wlen(m,bxmin)
1112      xmax=wlen(m,bxmax)
1113      C
1114      c makes sure there are enough pts to make a decent looking plot
1115      write(6,"j bxmax,bxmin",bxmax,bxmin)
1116      write(6,"j xmin,xmax",xmin,xmax)
1117      if((bxmax-bxmin).lt.6) then
1118          write(6,"Not enough wavelengths for a good plot'
1119          write(6,")
1120      if(bxmin.eq.1.and.bxmax.eq.nwlen(m)) goto 50
1121      goto 1115
1122      endif
1123      C
1124      c do the same thing as the x-axis
1125      if(ymin.eq.ymax) then
1126          iymn=1
1127          iymax=nang(m)
1128      C
1129      else
1130          iymn=1
1131      1164  if(ymin.gt.ang(m,iymn).and.iymn.lt.nang(m)) then
1132          iymn=iymn+1

```

## Appendix V

```

1133      goto 1164
1134      endif
1135      iymax=nang(m)
1136 1166 if(iymax.lt.angl(m,iymax).and.iymax.gt.1) then
1137      iymax=iymax-1
1138      goto 1166
1139      endif
1140      endif
1141      if(iymin.gt.1) iymin=iymin-1
1142      if(iymax.lt.nang(m)) iymax=iymax+1
1143      ymin=angl(m,iymin)
1144      ymax=angl(m,iymax)
1145      C
1146      c makes sure there are enough pts to make a decent looking plot
1147      if((iymax-iymin).lt.6) then
1148          write(6,*)"Not enough incident angles for a good plot"
1149          write(6,*)
1150      if(iymin.eq.1.and.iymax.eq.nang(m)) goto 50
1151      goto 1120
1152      endif
1153      C
1154      c find the min and max of Fij then set up limits
1155      call mg3d(idx,bxmin,bxmax,iymin,iymax,zmin,zmax)
1156      write(6,*)"xmin,zmax",xmin,zmax
1157      call setlim(xmin,xmax,x1,x2,10.)
1158      call setlim(ymin,ymax,y1,y2,10.)
1159      call setlim(zmin,zmax,z1,z2,10.)
1160      write(6,*)"x1,x2,y1,y2,z1,z2"
1161      write(6,*)"x1,x2,y1,y2,z1,z2"
1162      C
1163      1150 call stplot
1164      c draw center caption
1165      call head3d(1,matnm(m),autnm(icode(m)),hmq(m),sigs(m)
1166      *      ,elt,n1,n2)
1167      if(n2.ne.0) then
1168          call lgbase(matnm(n2),autnm(icode(n2)),hmq(n2),sigs(n2))
1169      endif
1170      c set up plot area, draw the axes, and then do the plot
1171      call prtn3d(vx,vy,vz)
1172      call axes3d(x1,x2,y1,y2,z1,z2)
1173      call dot3d(bxmin,bxmax-bxmin+1,iymin,iymax-iymin+1,
1174      *      angl,wini,m,idx,fwrk)
1175      call enplot
1176      C
1177      c Allow various changes without going to then main menu
1178      1180 write(6,1)
1179      write(6,*)"Enter 1 for another view point"
1180      write(6,*)"Enter 2 for a different mtx element"
1181      write(6,*)"Enter 0 to return to the main menu"
1182      read(5,*)i
1183      if(i.eq.1) then
1184          iflag2=1
1185          goto 1125
1186      else if(i.eq.2) then

```

## Appendix V

```

1187      iflag1-1
1188      goto 1101
1189      else if(l.eq.0) then
1190          goto 50
1191      endif
1192      goto 1180
1193      C
1194      C----- Option 43 -----
1195      C
1196      C This section does contour plotting
1197      c make sure there is a surface to use
1198      1200 continue
1199      if(nsurf.lt.1) then
1200          write(6,*)'No surfaces in the to use list'
1201          write(6,*)'Use main menu option 21 and try again'
1202          call rcont
1203          goto 50
1204      endif
1205      C
1206      c get mtx element data
1207      1201 write(6,1)
1208          write(6,*)'Enter the mtx element position you want using'
1209          write(6,*)'the matrix notation'
1210          write(6,*)'e.g. if you want F11, F22, or F34 then enter//'
1211          *      ' 11 or 22 or 34'
1212          write(6,*)'Enter 0 to return to the main menu'
1213          read(5,*)i
1214          if(i.eq.0) goto 50
1215          elt=i-6*(i/10)-4
1216          if(elt.lt.1.or.elt.gt.16) goto 1201
1217      C
1218      c print what the plot will be of
1219      1202 continue
1220          write(6,1)
1221          write(6,12110)
1222      12110 format(' These plots will be for mtx. element F',i2)
1223          write(6,*)
1224      C
1225      c Use no more than the 1st four elements of surfaces list
1226      if(nsurf.gt.4) then
1227          numplt=4
1228      else
1229          numplt=nsurf
1230      endif
1231      C
1232          write(6,12119)'Assign # ',' Material ',' Corr. fac. ',
1233          *      ' mean sq hgt ',' mean sq slope'
1234      12119 format(a8,2x,a13,2x,a13,2x,a13,2x,a13)
1235      do 1203 i=1,numplt
1236          m=surf(i)
1237          write(6,12120)m,matnm(m),autonm(acode(m)),hmq(m),sigx(m)
1238      12120 format(18,2x,a13,2x,a13,2x,f13.3,2x,f13.3)
1239      1203 continue
1240          write(6,*)

```

## Appendix V

```

1241  C
1242  c get wavelength and inc angle range
1243  1215 write(6,1)
1244      write(6,"Enter the range on wavelength (min max)"
1245      read(5,"xmin,xmax
1246      if(xmin.lt.0.0.or.xmin.ge.xmax) goto 1215
1247  C
1248  1220 write(6,"Enter the range on incident angle (min max)"
1249      read(5,"ymin,ymax
1250      if(ymin.lt.0.0.or.ymax.gt.90.0.or.ymin.ge.ymax) goto 1220
1251  C
1252  c let the user out if he sees a mistake
1253  1228 write(6,"Enter 1 if this is correct"
1254      write(6,"Enter 0 to return to the main menu"
1255      read(5,"i)
1256      if(i.eq.1) goto 1229
1257      if(i.eq.0) goto 50
1258      goto 1228
1259  C
1260  c read in the data for all the plots
1261  c find the base grid bounded by the user's range
1262  c find xmax and xmin over all the plots
1263  1229 xmax=-1.e36
1264      xmin=-1.e36
1265      do 1269 i=1,numpit
1266          m=surf(i)
1267          j=frum(i)
1268          frum(i)=elt
1269          call freads(m,1,frum,frame(m),rcode(m),nskip(m))
1270          frum(i)=j
1271          call ffind(m,elt,idx)
1272          write(6,"ym,elt,idx",m,elt,idx
1273          bxmin=1
1274  1260  if(xmin.gt.wlen(m,bxmin).and.bxmin.lt.nwlen(m)) then
1275          bxmin=bxmin+1
1276          goto 1260
1277          endif
1278          bmax=nwlen(m)
1279  1262  if(xmax.lt.wlen(m,bmax).and.bmax.gt.1) then
1280          bmax=bmax-1
1281          goto 1262
1282          endif
1283  C
1284  c contour plots look bad without a lot of base points
1285      write(6,"bmax,bxmin",bmax,bxmin
1286      write(6,"xmin,xmax",xmin,xmax
1287      if((bmax-bxmin).lt.10) then
1288          write(6,"Not enough wavelengths for a good plot"
1289          write(6,"
1290          if(bxmin.eq.1.and.bmax.eq.nwlen(m)) goto 50
1291          goto 1215
1292      endif
1293  C
1294      ymax=1

```

# Appendix V

```

1295 1264 if(ymin.gt.ang(m,lymin).and.iymin.lt.nang(m)) then
1296     lymin=lymin+1
1297     goto 1264
1298 endif
1299 lymin=nang(m)
1300 1266 if(ymax.lt.ang(m,lymax).and.iymax.gt.1) then
1301     lymin=lymin-1
1302     goto 1266
1303 endif
1304 C
1305 if((lymax-lymin).lt.10) then
1306     write(6,*)'Not enough incident angles for a good plot'
1307     write(6,*)
1308     if(lymin.eq.1.and.lymax.eq.nang(m)) goto 50
1309     goto 1220
1310 endif
1311 C
1312 c this call gets min and max for one plot only
1313 call mg3d(kb,bmin,bmax,lymin,iymax,zmin2,zmax2)
1314 write(6,*)'zmin2,zmax2',zmin2,zmax2
1315 if(zmin2.lt.zmin) zmin=zmin2
1316 if(zmax2.gt.zmax) zmax=zmax2
1317 c save base grid ranges for each plot separately
1318 index(i)=idx
1319 bmin(i)=bmin
1320 bmax(i)=bmax
1321 lymin(i)=lymin
1322 iymax(i)=iymax
1323 1269 continue
1324 C
1325 c set up limits and decide what 9 common contours to use on all the plots
1326 call setlim(xmin,xmax,x1,x2,10.)
1327 call setlim(ymin,ymax,y1,y2,10.)
1328 call setlim(zmin,zmax,z1,z2,10.)
1329 write(6,*)'x1,x2,y1,y2,z1,z2'
1330 write(6,*)'x1,x2,y1,y2,z1,z2'
1331 delz=(z2-z1)/10.
1332 do 1270 i=1,9
1333     lv(i)=delz*(i)+z1
1334 1270 continue
1335 C
1336 call stplot
1337 c same heading as 3d plots
1338 call head3d(neurf,matnm(m),autnm(acode(m)),hmaq(m),sigs(m)
1339     ,ek,n1,n2)
1340 c put a legend for the contour lines in
1341 call lgcn(lvl)
1342 if(n2.ne.0) then
1343     call lgbase(matnm(n2),autnm(acode(n2)),hmaq(n2),sigs(n2))
1344 endif
1345 c do each plot in sequence but use a common set of contour levels
1346 do 1280 i=1,numpit
1347     m=par(i)
1348     call prn2d(numpit,i)

```

## Appendix V

```

1349      call axesec(x1,x2,y1,y2,nsurf,matnm(m),autonm(scode(m))
1350      ,hnaq(m),siga(m))
1351      call dotm(xmxn(i),txmx(i)-txmn(i)+1,iymn(i),lymx(i)-iymn(i)+1,
1352      *      angl,wlenl,m,index(i),lvl,z1,z2)
1353      call engr2d
1354 1280 continue
1355      call enplot
1356      goto 50
1357 C
1358 C In the main segment of this program no calls have been made directly
1359 C to DISPLA. This means another plotting library can be used if the
1360 C subroutines are rewritten to perform the same tasks as described
1361 C after this. Those subroutines with direct calls to DISPLA are noted
1362 C
1363      end
1364 C***** End of MAIN *****
1365 C
1366 C
1367 C***** Subroutine rcont *****
1368 C
1369      subroutine rcont
1370 c let the user look at something and prompt him to continue
1371      write(6,*)
1372      write(6,*)'enter <RETURN> to continue'
1373      read(5,*)
1374      write(6,*)
1375      return
1376      end
1377 C
1378 C
1379 C***** Subroutines freesd, fnorm2, fnorm1, finit, fupdt *****
1380 C
1381 C
1382 C This routine has NO Displa calls
1383 C
1384 C These subroutines along with find make up the data management section
1385 C Since the data files can be very long this section stores only the
1386 C requested information. Instead of storing all Pij for each surface
1387 C only selected mti elts are stored.
1388 C Storage is in array fsi(index,lang,lwin)
1389 C index is the number assigned by freesd and can be determined for a
1390 C given surface assignment number and a given mti elt by find
1391 C Up to maxdet indices are available. If fsi fills up then the oldest
1392 C index is removed to make room for the next one.
1393 C For each index the surface assignment number, the mti elt num, and age are saved
1394 C in arrays surf, elt, and age1.
1395 C
1396      subroutine freesd(m,nelt,fnum,fname,scode,nakip)
1397      Integer maxplt,maxcur,maxang,maxwin,maxaur,maxdet
1398      Integer dcode,scode2,dnakip,nakip2
1399      parameter(maxplt=4,maxcur=7,maxang=25)
1400      parameter(maxwin=71,maxaur=10,maxdet=20)
1401      Integer m,index(maxplt),nelt
1402      Integer fnum(maxplt),elt(maxplt),scode,nakip

```

## Appendix V

```

1403      Integer surfi(maxdat),elti(maxdat),agei(maxdat)
1404      Integer i,j,date,oldest,idx
1405      Integer n1,n2,dn1,dn2,mtemp
1406      Character*13 fname,dname,aname2
1407      Common/four/surfi,elti
1408      save
1409      C
1410      C
1411      c if a base material is desired its values are stored with assignment # 99
1412      c so the info can be used for subtractions elsewhere
1413      if(n2.ne.0) then
1414          flag1=1
1415          mtemp=m
1416          m=99
1417      endif
1418      2112 continue
1419      C
1420      k=0
1421      c every access increase date so that age of an index can be determined
1422      date=date+1
1423      c see if requested read already exists
1424      do 2110 i=1,nelt
1425          idx=0
1426          do 2115 j=1,maxdat
1427              if(surfi().eq.m.and.elti().eq.fnum(i)) idx=j
1428          2115 continue
1429      C
1430      c only read if you have to and put data into the oldest index
1431      c also k mtr elements can be read in one shot so
1432      c more than one index may be available
1433      if(idx.eq.0) then
1434          k=k+1
1435          elt(k)=fnum(i)
1436          oldest=10000
1437          do 2130 j=1,maxdat
1438              if(agei().lt.oldest) then
1439                  oldest=agei(j)
1440                  index(k)=j
1441              endif
1442          2130 continue
1443      c store indexing information and date it
1444          surfi(index(k))=m
1445          elti(index(k))=fnum(i)
1446          agei(index(k))=date
1447      endif
1448      2110 continue
1449      C
1450      c k is then number of mtr elts to read
1451      if(k.gt.0) then
1452          if(flag1.eq.1) then
1453              c if reading base data don't do the base normalization because you'd get 0
1454              call readf(index,elt,k,fname2,acode2,naktp2,n1,0)
1455          else
1456              c otherwise do the reading with the appropriate normalizations

```

## Appendix V

```

1457      call readsf(index,elt,k,fname,acode,nskip,n1,n2)
1458      endif
1459      endif
1460      C
1461      c if base data read in then go back and do the data for plotting
1462      if(iflag1.eq.1) then
1463          iflag1=0
1464          m=ntemp
1465          goto 2112
1466      endif
1467      return
1468      C
1469      C
1470      C this entry accepts new base normalization info and saves it for freads
1471      entry fnorm2(dn2,dfname,dacode,dnskip)
1472      n2=dn2
1473      fname2=dfname
1474      acode2=dacode
1475      nskip2=dnskip
1476      return
1477      C
1478      C
1479      C this entry accepts new flag for P11 normalization and saves it
1480      entry fnorm1(dn1)
1481      n1=dn1
1482      return
1483      C
1484      C
1485      C this entry is used to throw away all old information when normalizations
1486      C have been changed or at startup to clear the memory
1487      entry finit
1488      date=0
1489      do 2140 i=1,maxdat
1490          surf(i)=0
1491          etlt(i)=0
1492          age(i)=0
1493      2140 continue
1494      return
1495      C
1496      C
1497      C this entry is used to throw away information pertaining to a given
1498      C assignment number that has been deleted.
1499      entry fupdt(m)
1500      do 2150 i=1,maxdat
1501          if(surf(i).eq.m) then
1502              surf(i)=0
1503              etlt(i)=0
1504              age(i)=0
1505          endif
1506      2150 continue
1507      return
1508      end
1509      C
1510      C

```

## Appendix V

```

1511 C***** Subroutine ffind *****
1512 C
1513 C This routine has NO Displa calls
1514 C
1515 C This routine interrogates indexing info to find the index for a given
1516 C assignment # and mtr elt combination
1517     subroutine ffind(m,elt,index)
1518     integer maxplt,maxcur,maxang,maxwin,maxsur,maxdat
1519     parameter(maxplt=4,maxcur=7,maxang=25)
1520     parameter(maxwin=71,maxsur=10,maxdat=20)
1521     integer m,elt,index,k
1522     integer surfl(maxdat),eltl(maxdat)
1523     common/four/surfl,eltl
1524 C
1525 c find match to requested data
1526     do 2120 k=1,maxdat
1527         if(surfl(k).eq.m.and.eltl(k).eq.elt) then
1528             index=k
1529             return
1530         endif
1531     2120 continue
1532     index=0
1533     return
1534 end
1535 C
1536 C
1537 C***** Subroutine readaf *****
1538 C
1539 C This routine has NO Displa calls
1540 C
1541 C This subroutine reads in data into array fst(...) using the given
1542 C index list, elt list, filename, correlation code, and normalizations provided
1543 C Two reading loops are provided to read in experimental and theoretical data
1544 C
1545     subroutine readaf(index,elt,nelt,fname,acode,nakip,n1,n2)
1546     integer maxplt,maxcur,maxang,maxwin,maxsur,maxdat
1547     parameter(maxplt=4,maxcur=7,maxang=25)
1548     parameter(maxwin=71,maxsur=10,maxdat=20)
1549     integer index(maxplt),elt(maxplt),nelt
1550     integer l,lang,'wlen,nakip
1551     integer nh,na,nwlen,nang
1552     integer acode,n1,n2,idx2
1553     real fst(maxdat,maxwin,maxang)
1554     real a,b,c,d,e,g,dum(16),q(3)
1555     character*13 fname
1556 C
1557     common/one/fst
1558 C
1559 c read in the header information
1560     open(11,file=fname,status='old')
1561     read(11,'end=911)
1562     read(11,'end=911)nh,na,nwlen,nang
1563     read(11,'end=911)(x,i=1,nh)
1564     read(11,'end=911)(x,i=1,na)

```

# Appendix V

```

1565      read(11,*,end=911)(x,i-1,nwlen)
1566      read(11,*,end=911)(x,i-1,nang)
1567 C
1568 c skip over the proper number of data lines for theoretical data
1569   if (nakip.gt.0) then
1570     do 2010 i-1,nakip
1571       read(11,*,end=911)
1572 2010 continue
1573     endif
1574 C
1575 c read in exp data
1576 c read extra lines for experimental data
1577 c read in 16 values on 3 lines for each iwlen and iang
1578 c save only the desired ones into fst(...)
1579   if(acode.eq.8) then
1580     do 2011 i-1,nakip/2
1581       read(11,*,end=911)
1582 2011 continue
1583     do 2020 iwlen-1,nwlen
1584       do 2020 iang-1,nang
1585         read(11,*,end=911)(dum(i),i-1,16)
1586       do 2020 i-1,nelt
1587         fst(index(i),iwlen,iang)-dum(elt(i))
1588 2020 continue
1589 C
1590 c read in theoretical data.
1591 c for RETRO produced files these 8 elts are assumed 0.
1592 c read in 3 Q values and 6 unique elts for each iwlen and iang.
1593 c for F11 normalization the Q values cancel out.
1594 c a Q<0 is assumed --> 0.
1595 c do F11 normalization.
1596 c combine Q's back into the elts.
1597 c save only desired mtr elts in fst(...).
1598   else
1599     dum(3)=0.
1600     dum(4)=0.
1601     dum(7)=0.
1602     dum(8)=0.
1603     dum(9)=0.
1604     dum(10)=0.
1605     dum(13)=0.
1606     dum(14)=0.
1607 C
1608     do 2040 iwlen-1,nwlen
1609       do 2040 iang-1,nang
1610         read(11,*,end=911)q(1),q(2),q(3)
1611 C
1612         if(n1.eq.1) then
1613           q(1)=1.
1614           q(2)=1.
1615           q(3)=1.
1616         else
1617           if(q(1).lt.0.) q(1)=0.
1618           if(q(2).lt.0.) q(2)=0.

```

# Appendix V

```

1619      W(q(3),k,0.) q(3)=0.
1620      endif
1621      read(11,*,end=911)a,b,c,d,e,g
1622      C
1623      if(n1.eq.1) then
1624          b=b/a
1625          c=c/a
1626          d=d/a
1627          e=e/a
1628          g=g/a
1629          a=1.
1630      endif
1631      C
1632      dum(1)=a
1633      dum(2)=b
1634      dum(3)=b
1635      dum(6)=c
1636      dum(11)=d
1637      dum(12)=e
1638      dum(15)=e
1639      dum(16)=g
1640      C
1641      do 2040 i=1,nelt
1642          if(acode.lt.4) then
1643              fst(index(i),iwlen,iang)=q(acode)*dum(elt(i))
1644          else if(acode.lt.7) then
1645              fst(index(i),iwlen,iang)=q(acode-3)
1646          else
1647              fst(index(i),iwlen,iang)=dum(elt(i))
1648          endif
1649      2040 continue
1650      endif
1651      C
1652      c do base normalisations
1653      c find where the base data is stored
1654      c go through and do subtractions
1655      if(n2.ne.0) then
1656          do 2050 i=1,nelt
1657              call ffind(99,elt(i),idx2)
1658              do 2050 j=1,nwlen
1659                  do 2050 k=1,nang
1660                      fst(index(i),j,k)=(fst(index(i),j,k)-fst(idx2,j,k))/fst(index2,j,k)
1661      2050 continue
1662      endif
1663      C
1664      c close and rewind file
1665      close(11)
1666      return
1667      C
1668      c print out an error in reading data but don't kill program
1669      911 write(6,"not enough data on unit 11")
1670      write(6,"returning to program anyway")
1671      write(6,"proceed at your own risk")
1672      call rcost

```

## Appendix V

```

1673      return
1674      end
1675      C
1676      C
1677      C***** Subroutines stplot, stplt2, stplt3, stplt4, stplt5, stplt6 *****
1678      C
1679      C This routine has DISPLA routines
1680      C
1681      C stplot nominates the terminal, sets page, and many plot features
1682      C stplt2, stplt3, stplt4, stplt5, and stplt6 pass info to stplot
1683      subroutine stplot
1684      integer lterm,dterm,ntickx,dntickx,nticky,dnticky,dndraw,ndraw
1685      integer lhide,dhide
1686      real xac,yac,xscale,ycale
1687      save
1688      C
1689      c nominate 1 of three terminal types
1690      if(lterm.eq.1) then
1691        call tk41(4115)
1692      else if(lterm.eq.2) then
1693        call tk41(4107)
1694      else if(lterm.eq.3) then
1695        call vt240
1696      endif
1697      C
1698      c at a new page, send displa output to fort.7, set up page, scale for headings
1699      c and legends, setup delimiters for embedded commands for messag.
1700      c set up macros to do mean square height and mean square slope symbols
1701      c set up tick marks, curve thickness, and set hidden line removal status
1702      c
1703      call bgnp(0)
1704      call setdev(7,7)
1705      call page(12.*xac,10.5*yac)
1706      call bscale(xac,yac)
1707      call nochet
1708      call coemptx
1709      call base1('STANDARD')
1710      call embas1('INSTRUCTION')
1711      call s1use('M11')//char(37)//'(M1)H(EH.5)2(EXHXM11)//
1712      * char(38)//'(MX',29)
1713      call s2use('M7)5(MXH.5L.25)5(B1LXHXEH.5)2(EXHXMX',33)
1714      call xticks(ntickx)
1715      call yticks(nticky)
1716      call thicrv(ndraw)
1717      if(lhide.eq.0) call nohide
1718      return
1719      C
1720      C
1721      c pass in terminal code
1722      entry stplt2(dterm)
1723      lterm=dterm
1724      return
1725      C
1726      C

```

## Appendix V

```

1727  c pass in tick mark density
1728      entry stpt3(dntickx,dnticky)
1729      ntickx-dntickx
1730      nticky-dnticky
1731      return
1732  C
1733  C
1734  c pass in plot scaling
1735      entry stpt4(xscale,yscale)
1736      xsc-xscale
1737      ysc-yscale
1738      return
1739  C
1740  C
1741  c pass in curve thickness parameter
1742      entry stpt5(dndraw)
1743      ndraw-dndraw
1744      return
1745  C
1746  C
1747  c pass in hidden line removal status
1748      entry stpt6(dihide)
1749      ihide-dihide
1750      return
1751      end
1752  C
1753  C
1754  C===== Subroutine enplot =====
1755  C
1756  C This routine has DSSPLA calls
1757  C
1758  C this routine end the current page layout
1759      subroutine enplot
1760      call endpl(0)
1761      close(7)
1762      call rcont
1763      return
1764      end
1765  C
1766  C
1767  C===== Subroutine head2d =====
1768  C
1769  C This routine has DSSPLA calls
1770  C
1771  C This routine puts the center heading on the page
1772  C M1 and M2 correspond to Mflag1 and Mflag2 in option 41
1773  C
1774      subroutine head2d(numpit,matnm,autonm,hmaq,slgs,
1775      *               wlen,ang,M1,M2,ptype)
1776      Integer numpit,M1,M2,ptype
1777      Real hmaq,slgs,wlen,ang
1778      Character*13 matnm
1779      Character*13 autonm
1780      Character*1 schar

```

# Appendix V

```

1781 C
1782 C
1783 if(numplt.eq.1) then
1784     achar=' '
1785 else
1786     achar='S'
1787 endif
1788 C
1789 c set up plot area and location to be ready for messag commands
1790 call complx
1791 call physort(0.,0.)
1792 call area2d(10.5,9.0)
1793 C
1794 c put in 1st and 2nd lines using the self centering headin routines
1795 c 1st line is main title
1796 call headin('BACKSCATTER MUELLER MATRIX ELEMENT'//achar,35,1.6,4)
1797 c 2nd line is the material used line
1798 c indicating a comparison
1799 if(lf1.eq.0) then
1800     call headin('Comparison of Materials',100,1.3,4)
1801 c or the actual one used
1802 else
1803     call headin(matnm/' '//autnm/'',100,1.3,4)
1804 endif
1805 call headin(' ',100,1.3,4)
1806 call headin(' ',100,1.3,4)
1807 C
1808 c put in a line describing rough parameters for this page
1809 c either a statement of comparison or
1810 if(lf1.eq.1) then
1811     call messag('(H1.3)Comparison of Roughness Parameters',
1812     *      100,3.0,9.5)
1813 c the actual values used
1814 else
1815     call messag('(H1.3)MXZ1) - ',100,3.7,9.5)
1816     call realno(hmeq,104,'ABUT','ABUT')
1817     call messag('(HXM7)M(M1)M(EH.5)2(EXHXM)'//',100,'ABUT','ABUT')
1818     call messag('(H1.3) (Z2) - (HXM)'//',100,'ABUT','ABUT')
1819     call realno(aigs,104,'ABUT','ABUT')
1820 endif
1821 C
1822 c the 4th line provides info about the fixed inc ang or fixed wavelength
1823 if(lf2.eq.1) then
1824     call messag('(H1.3)Comparison of Incident Angles',100,3.5,9.2)
1825 c
1826 else if(lf2.eq.0) then
1827     call messag('(H1.3)Comparison of Wavelengths',100,3.7,9.2)
1828 c
1829 c put in the inc ang if only one used
1830 else
1831     if(ptype.lt.3) then
1832         call messag('(H1.3)Q(M1H.5)Q(LXHXM)'//',100,4.8,9.2)
1833         call realno(ang,104,'ABUT','ABUT')
1834         call messag('(M1EH.5)Q(MXEXH)'//',100,'ABUT','ABUT')

```

# Appendix V

```

1835 C
1836 c put in the wavelength if only one used
1837     else
1838         call messag('H1.3M7)L(M1H.5L)Q(LXHXMX) - ',100,4.8,9.2)
1839         call realno(wlen,104,'ABUT','ABUT')
1840         call messag('H.8M7)M(M1)M(HXMX)''',100,'ABUT','ABUT')
1841     endif
1842 endif
1843 c reset so that next routine can have its own sub plot area
1844     call endgr(0)
1845     return
1846 end
1847 C
1848 C
1849 C***** Subroutine lgrmat *****
1850 C
1851 C This routine has DISPLA calls
1852 C
1853 C This routine prints the list of materials/corr. fnc used in a comparison
1854 C
1855     subroutine lgrmat(numcur,matnm,maxsur,surf,acode,autnm)
1856     integer numcur,i,surf(numcur),maxsur,acode(maxsur)
1857     Character*13 matnm(maxsur)
1858     Character*13 autnm(8)
1859 C
1860 c define a subplot area and display the info, changing color or line type
1861     call physar(0.,0.)
1862     call area2d(10.5,9.0)
1863     call messag('H1.3)Materials'',100,9.9,9.85)
1864     do 3010 i=1,numcur
1865         call lines(i)
1866         call strtpt(8.6,9.9-.2*i)
1867         call connpt(9.2,9.9-.2*i)
1868         call reset('DASH')
1869         call messag(matnm(surf(i))/' '//autnm(acode(surf(i)))//
1870     *      ' ',100,9.3,9.8-.2*i)
1871     3010 continue
1872     call reset('SETCLR')
1873 c end this subplot
1874     call endgr(0)
1875     return
1876 end
1877 C
1878 C
1879 C***** Subroutine lgrul *****
1880 C
1881 C This routine has DISPLA calls
1882 C
1883 C This routine prints the roughness scales if a comparison is being made
1884 C
1885     subroutine lgrul(numcur,hmaq,sigs,maxsur,surf)
1886     integer numcur,maxsur,surf(numcur),i
1887     Real hmaq(maxsur),sigs(maxsur)
1888 C

```

## Appendix V

```

1889 c set up subplot area and display info while changing color or line type
1890     call physar(0.,0.)
1891     call area2d(10.5,9.0)
1892     call messag('(H1.3Z1), (Z2)*',100,10.2,10.15)
1893     do 3020 i=1,numcur
1894         call lines(i)
1895         call strpt(9.3,10.2-.2*i)
1896         call connpt(9.9,10.2-.2*i)
1897         call reset('DASH')
1898         call realno(hmaq(surf(i)),103,10.,10.1-.2*i)
1899         call messag('(H.8M7)M(M1)M(MXHXEH.5)2(EXHX), ~',
1900             *      100,'ABUT','ABUT')
1901         call realno(sigs(surf(i)),104,'ABUT','ABUT')
1902     3020 continue
1903     call reset('SETCLR')
1904     call endgr(0)
1905     return
1906     end
1907 C
1908 C
1909 C***** Subroutine lgwin *****
1910 C
1911 C
1912 C This routine has DISPLA calls
1913 C
1914 C This routine prints the multiple fixed wavelengths if needed
1915 C Very similar format to other legend makers
1916 C
1917     subroutine lgwin(numcur,wlenfx)
1918     Integer numcur,i
1919     Real wlenfx(numcur)
1920 C
1921     call physar(0.,0.)
1922     call area2d(10.5,9.0)
1923     call messag('(H1.3M7)L(M1H.5L)(LXHX)*',100,10.2,10.15)
1924     do 3030 i=1,numcur
1925         call lines(i)
1926         call strpt(9.3,10.2-.2*i)
1927         call connpt(9.9,10.2-.2*i)
1928         call reset('DASH')
1929         call realno(wlenfx(i),104,10.,10.1-.2*i)
1930         call messag('(H.8M7)M(M1)M(HXMX)*',100,'ABUT','ABUT')
1931     3030 continue
1932     call reset('SETCLR')
1933     call endgr(0)
1934     return
1935     end
1936 C
1937 C
1938 C***** Subroutine lgang *****
1939 C
1940 C This routine has DISPLA calls
1941 C
1942 C This routine prints out the multiple fixed inc. angles if comparing

```

# Appendix V

```

1943 C very similar to other legend makers
1944 C
1945 subroutine lgang(numcur,angfx)
1946 integer numcur,i
1947 Real angfx(numcur)
1948 C
1949 call physarr(0.,0.)
1950 call area2d(10.5,9.0)
1951 call messag('H1.3M7)Q(M1H.5L)O(LXHX)*',100,10.2,10.15)
1952 do 3040 i=1,numcur
1953 call liness(i)
1954 call strip(9.3,10.2-.2*i)
1955 call connpt(9.9,10.2-.2*i)
1956 call reset('DASH')
1957 call resino(angfx(i),104,10.,10.1-.2*i)
1958 call messag('M1EH.5)O(MXEXHX)*',100,'ABUT','ABUT')
1959 3040 continue
1960 call reset('SETCLR')
1961 call endgr(0)
1962 return
1963 end
1964 C
1965 C
1966 C***** Subroutine liness, setlin *****
1967 C
1968 subroutine liness(i)
1969 integer i,j,ifcol,difcol
1970 Real r1(4),r2(6),r3(3),r4(8)
1971 Data r1/11.,3.,3.,3./
1972 Data r2/8.,3.,2.,2.,2.,3./
1973 Data r3/5.,10.,5./
1974 Data r4/4.,3.,2.,3.,2.,3.,4.,3./
1975 C
1976 if(ifcol.eq.1) then
1977 j=1+(i/7)*7
1978 if(j.eq.1) call setclr('BLUE')
1979 if(j.eq.2) call setclr('GREEN')
1980 if(j.eq.4) call setclr('CYAN')
1981 if(j.eq.3) call setclr('RED')
1982 if(j.eq.5) call setclr('MAGENTA')
1983 if(j.eq.6) call reset('SETCLR')
1984 if(j.eq.0) call setclr('YELLOW')
1985 return
1986 else
1987 C
1988 j=1+(i/6)*6
1989 if(j.eq.0) call dash
1990 if(j.eq.1) call reset('DASH')
1991 if(j.eq.2) call mrsacd(4,4,r1)
1992 if(j.eq.3) call mrsacd(4,6,r2)
1993 if(j.eq.4) call mrsacd(4,3,r3)
1994 if(j.eq.5) call mrsacd(4,8,r4)
1995 return
1996 endif

```

## Appendix V

```

1997  C
1998  C
1999  entry setlin(dficol)
2000  ificol=dficol
2001  return
2002  end
2003  C
2004  C
2005  C***** Subroutine setlim *****
2006  C
2007  C This routine has NO display calls
2008  C
2009  C This routine returns ymin and ymax which should be better limits for
2010  C setting up an axis than the absolute values y1 and y2 passed in.
2011  C round controls how many sig figs ymin and ymax will contain.
2012  C round=10 -> 2 sig figs, round=1000 -> 4 sig figs
2013  C
2014  subroutine setlim(y1,y2,ymin,ymax,round)
2015  Integer I1,I2,Itemp
2016  Real y1,y2,ymin,ymax,round
2017  Real y1p,y2p,y1pp,y2pp,ytemp
2018  C
2019  c find true min and max if values switched around
2020  If(y1.gt.y2) then
2021    ytemp=y1
2022    y1=y2
2023    y2=ytemp
2024  endif
2025  C
2026  c special cases exist if y1-y2
2027  If(y1.eq.y2) then
2028    If(y1.eq.0.) then
2029      ymin=-1.
2030      ymax=1.
2031    else If(y1.lt.0.) then
2032      ymin=1.5*y1
2033      ymax=.5*y1
2034    else
2035      ymin=.5*y1
2036      ymax=1.5*y1
2037    endif
2038    return
2039  else
2040  C
2041  c otherwise break the numbers down
2042  If(y1.eq.0.) y1=1.e-36
2043  If(y2.eq.0.) y2=1.e-36
2044  I1=fix((alog10(abs(y1))+40)-40)
2045  I2=fix((alog10(abs(y2))+40)-40)
2046  If(I1.gt.I2) then
2047    Itemp=I1
2048    I1=I2
2049    I2=Itemp
2050  endif

```

## Appendix V

```

2051  C
2052  c y1p and y2p have a maximum of 1 place left of the dec pt except
2053  c in the special circumstance when that place is not a true sig fig
2054  3500  y1p=y1/(10.**float(l2))
2055        y2p=y2/(10.**float(l2))
2056        if((ifix(y1p).eq.0.and.ifix(y2p).eq.1).or.
2057        *   (ifix(y1p).eq.-1.and.ifix(y2p).eq.0)) then
2058            l2=l2-1
2059            goto 3500
2060        endif
2061  C
2062  c round the numbers off on a common basis
2063  c let plot limits fall inside absolute values just slightly if need be
2064        y1pp=ifix(y1p*round)/round
2065        if(y1.lt.0.0.and.y1pp.gt.y1p+.01/round) y1pp=y1pp-(1./round)
2066        y2pp=ifix(y2p*round)/round
2067        if(y2.gt.0.0.and.y2pp.lt.y2p-.01/round) y2pp=y2pp+(1./round)
2068  C
2069  c remove the normalization factor
2070        ymin=y1pp*(10.**l2)
2071        ymax=y2pp*(10.**l2)
2072        endif
2073        return
2074        end
2075  C
2076  C
2077  C***** Subroutine curve *****
2078  C
2079  C This routine has NO Displa calls
2080  C
2081  C this routine takes info out of data storage and puts it into arrays
2082  C for plotting. The data is stored in 2-d format, curve pulls the data out
2083  C as a fnc of wlen at a fixed inc ang or it pulls data out as a fnc of inc ang
2084  C at a fixed wavelength. The fixed inc ang or wavelength need not lie directly
2085  C on a grid because this routine linearly interpolates the between grids
2086  C
2087  C This routine fills curve numbers k1 to k2 of arrays xray and yray
2088  C either k1-1 and k2-numcur or k1-k2
2089  C
2090  subroutine curve(k1,k2,m,frum1,angl,wlen1,nang1,nwlen1,
2091  *               angfx,wlenfx,ptype,numpts,xray,yray)
2092  integer maxplt,maxcur,maxang,maxwin,maxsur,maxdat
2093  parameter (maxplt=4,maxcur=7,maxang=25)
2094  parameter (maxwin=71,maxsur=10,maxdat=20)
2095  integer k,k1,k2,m,frum1,nang1,nwlen1,numpts(maxcur),ptype
2096  real factl
2097  real angk(maxsur,maxang),wlenk(maxsur,maxwin)
2098  real angfx(maxcur),wlenfx(maxcur)
2099  real xray(maxcur,maxwin),yray(maxcur,maxwin)
2100  real fct(maxdat,maxwin,maxang)
2101  C
2102  common/one/fct
2103  C
2104  c get index of data storage

```

# Appendix V

```

2105      call ffind(m,frum1,idx)
2106 c get data as a function of wavelength for
2107      if(ptype.lt.3) then
2108          do 4000 k=k1,k2
2109 c determine fixed inc angle
2110          if(ptype.eq.1) then
2111              afix=angfx(k)
2112          else
2113              afix=angfx(1)
2114          endif
2115          numpts(k)=nwlen1
2116 c find what grid corresponds to the inc angle or find grids for interpolation
2117          j=1
2118          if(angl(m,j).gt.afx) goto 4040
2119          4020      j=j+1
2120          if(angl(m,j).gt.afx) goto 4030
2121          if(angl(m,j).eq.afx) goto 4040
2122          if(j.eq.nang1) goto 4040
2123          goto 4020
2124      C
2125 c do the interpolation to get data as a function of wavelength
2126      4030      j=j-1
2127          fact1=(afx-angl(m,j))/(angl(m,j+1)-angl(m,j))
2128          do 4050 i=1,nwlen1
2129              xray(k,i)=wlenl(m,i)
2130              yray(k,i)=(fst(idx,i,j+1)-fst(idx,i,j))*fact1+fst(idx,i,j)
2131          4050      continue
2132          goto 4000
2133      C
2134 c inc angle hit grid line exactly or requested inc ang fell outside grid range
2135      4040      do 4060 i=1,nwlen1
2136          xray(k,i)=wlenl(m,i)
2137          yray(k,i)=fst(idx,i,j)
2138      4060      continue
2139      4000      continue
2140      C
2141 c get data as a function of inc angle
2142      else
2143          do 4100 k=k1,k2
2144 c determine the fixed wavelength to use
2145          if(ptype.eq.3) then
2146              wfx=wlenfx(k)
2147          else
2148              wfx=wlenfx(1)
2149          endif
2150          numpts(k)=nang1
2151 c find the correct grid or grids to interpolate between
2152          j=1
2153          if(wlenl(m,j).gt.wfx) goto 4140
2154          4120      j=j+1
2155          if(wlenl(m,j).gt.wfx) goto 4130
2156          if(wlenl(m,j).eq.wfx) goto 4140
2157          if(j.eq.nwlen1) goto 4140
2158          goto 4120

```

## Appendix V

```

2159 C
2160 c do the interpolation
2161 4130 j=j-1
2162 facti=(wfix-wlenk(m,j))/(wlenk(m,j+1)-wlenk(m,j))
2163 do 4150 i=1,nangl
2164 xray(k,i)-angl(m,i)
2165 yray(k,i)-(fst(idx,j+1,i)-fst(idx,j,i))*facti+fst(idx,j,i)
2166 4150 continue
2167 goto 4100
2168 C
2169 c interpolation not needed
2170 4140 do 4160 i=1,nangl
2171 xray(k,i)-angl(m,i)
2172 yray(k,i)-fst(idx,j,i)
2173 4160 continue
2174 4100 continue
2175 endif
2176 return
2177 end
2178 C
2179 C
2180 C***** Subroutine rng2d *****
2181 C
2182 C This routine has NO displa calls
2183 C
2184 C this routine determines the absolute max and min of the y value
2185 C on a given 2-d plot for a single mtx element in the appropriate x-axis range
2186 C
2187 subroutine rng2d(numcur,xray,xmin,xmax,yray,numpts,
2188 * maxcur,maxwin,ymin,ymax)
2189 integer numcur,maxcur,maxwin,numpts(maxcur)
2190 integer i,j
2191 Real xray(maxcur,maxwin),yray(maxcur,maxwin),ymin,ymax,xmin,xmax
2192 C
2193 c normal min max search strategy
2194 ymin=-1.e37
2195 ymax=-1.e37
2196 do 4200 i=1,numcur
2197 do 4200 j=1,numpts(i)
2198 if(xray(i,j).ge.xmin.and.xray(i,j).le.xmax) then
2199 if(yray(i,j).gt.ymax) ymax=yray(i,j)
2200 if(yray(i,j).lt.ymin) ymin=yray(i,j)
2201 endif
2202 4200 continue
2203 return
2204 end
2205 C
2206 C
2207 C***** Subroutines prtn2d, sect2d *****
2208 C
2209 C This routine has DESSPLA calls
2210 C
2211 C This routine sets up the subplot area for 2-d and contour plots by dividing
2212 C the page into 1 or 4 section. For each plt number a different section

```

## Appendix V

```

2213 C is set up for use
2214 C
2215     subroutine prn2d(numplt,plt)
2216     integer numplt,plt
2217     Real xac,yac,xscale,yscale
2218 C
2219 c scaling must be done on physical origin
2220     if(numplt.eq.1) then
2221         call physcor(1.5*xac,1.5*yac)
2222     call height(.21)
2223     else
2224         if(plt.eq.1) call physcor(1.*xac,5.25*yac)
2225         if(plt.eq.2) call physcor(6.9*xac,5.25*yac)
2226         if(plt.eq.3) call physcor(1.*xac,1.*yac)
2227         if(plt.eq.4) call physcor(6.9*xac,1.*yac)
2228 c the scaling must be reset again for multiple subplots
2229     call bscale(xac*3.6.5,yac*4.5/9.)
2230     call height(.35)
2231     endif
2232     call area2d(9.5,6.5)
2233     return
2234 C
2235 C
2236 c this entry passes the proper scale in from the main program
2237     entry sec2d(xscale,yscale)
2238     xac=xscale
2239     yac=yscale
2240     return
2241     end
2242 C
2243 C
2244 C***** Subroutines axes2d, sgrd2d *****
2245 C
2246 C This routine has DISPLAY calls
2247 C
2248 C This routine draws and labels the axes for 2-d plots
2249 C
2250     subroutine axes2d(otype,fnum1,xmin,xmax,ymin,ymax,n1,n2)
2251     integer otype,fnum1,lower,icol,n1,n2
2252     integer ifgrd,difgrd,ngrdx,ngrdy,dngrdx,dngrdy
2253     Real xwin,xmax,x1,x2,ymin,ymax,y1,y2
2254 C
2255 c choose proper name for x-axis
2256     if(otype.lt.3) then
2257         call xname('F4)Wavelength in (M7)M(M1)M'',100)
2258     else
2259         call xname('F4)Angle of incidence in degrees'',100)
2260     endif
2261 C
2262     call yaxang(0.)
2263     call yname(' ',100)
2264 c put y-axis label on top of graph
2265     call message(' ',100,-3.6.7)
2266     call makef(fnum1,n1,n2)

```

# Appendix V

```

2267 C
2268 c choose limits and set up draw axes
2269 call setlim(xmin,xmax,x1,x2,10.)
2270 call setlim(ymin,ymax,y1,y2,2.)
2271 call slmptx
2272 call graf(x1,(x2-x1)/10.,x2,y1,(y2-y1)/5.,y2)
2273 call compbx
2274 c put in grid lines if indicated
2275 if (Mgrid.eq.1) then
2276 call dash
2277 call grid(ngrdx,ngrdy)
2278 call reset("DASH")
2279 endif
2280 return
2281 C
2282 C
2283 c pass in the correct number of grid lines for the x and y axes
2284 entry sgrid2d(dMgrid,dngrdx,dngrdy)
2285 Mgrid=dMgrid
2286 ngrdx=dngrdx
2287 ngdy=dngrdy
2288 return
2289 end
2290 C
2291 C
2292 C***** Subroutine makef *****
2293 C
2294 C This routine has DISPLA calls
2295 C
2296 C This routine prints Fij, Fij/F11, Fij/Fij(base)-1, or [Fij/F11]/[Fij/F11(base)]-1
2297 C dependent upon what normalizations are if effect
2298 C
2299 subroutine makef(elt,n1,n2)
2300 integer elt,n1,n2
2301 Character*1 cint(4)
2302 Data cint/'1','2','3','4'/
2303 irow=(elt-1)/4+1
2304 icol=elt-(irow-1)*4
2305 c print Fij
2306 call messag("F4)/P(H.5)"/cint(irow)/cint(icol)/"
2307 * ,100,'ABUT','ABUT')
2308 c print F11
2309 if(n1.eq.1) then
2310 call messag("F4)/P(H.5)11",100,'ABUT','ABUT')
2311 endif
2312 c print base Fij
2313 if(n2.ne.0) then
2314 call messag("F4)/P(H.5)"/cint(irow)/cint(icol)/
2315 * '(B2HXE.BH.6)base(B2)",100,'ABUT','ABUT')
2316 c print base F11
2317 if(n1.eq.1) then
2318 call messag("F4)/P(H.5)11(B2HXE.BH.6)base"
2319 * ,100,'ABUT','ABUT')
2320 endif

```

## Appendix V

```

2321      endif
2322      return
2323      end
2324      C
2325      C
2326      C***** Subroutine lgbase *****
2327      C
2328      C This routine has DISPLA calls
2329      C
2330      C This routine prints out the info about the base in upper left corner
2331      C It prints name, corr. fnc, height, and slop info
2332      C
2333      subroutine lgbase(matnum,autonm,hmaq,siga)
2334      Real hmaq,siga
2335      Character*13 matnum,autonm
2336      C
2337      c set up subplot area
2338      call physor(0.,0.)
2339      call area2d(11.5,10.5)
2340      call height(.14)
2341      call messag('Base material -',100.,2.9.7)
2342      call messag(matnum/'',100.,2.9.5)
2343      call messag(autonm/'',100.,2.9.3)
2344      call messag('Z1) - ',100.,2.9.1)
2345      call reatno(hmaq,104,'ABUT','ABUT')
2346      call messag('M7)M(M1)M(EH.5)2',100,'ABUT','ABUT')
2347      call messag('Z2) - ',100.,2.8.9)
2348      call reatno(siga,104,'ABUT','ABUT')
2349      call endgr(0)
2350      return
2351      end
2352      C
2353      C
2354      C***** Subroutine dot2d *****
2355      C
2356      C This routine has DISPLA calls
2357      C
2358      C This routine actually calls the curve drawing routines.
2359      C Variable dimensioning of xray1 and yray1 are used because curve()
2360      C only wants the correctly dimensioned outputs.
2361      C
2362      subroutine dot2d(icur,numpis,xray,yray,xray1,yray1,maxcur,maxwin)
2363      Integer icur,numpis,maxcur,maxwin
2364      Real xray(maxcur,maxwin),yray(maxcur,maxwin)
2365      Real xray1(numpis),yray1(numpis)
2366      C
2367      c copy into variably dimensioned arrays
2368      do 4400 i=1,numpis
2369          xray1(i)=xray(icur,i)
2370          yray1(i)=yray(icur,i)
2371      4400 continue
2372      c subplot set up by prin2d
2373      call grace(.1)
2374      call curve(xray1,yray1,numpis,0)

```

## Appendix V

```

2375      call reset("SETCLR")
2376      call reset("DASH")
2377      return
2378 C
2379 C
2380 c this call is used to finish off a mtr plot when all curves drawn
2381      entry engr2d
2382      call endgr(0)
2383      return
2384      end
2385 C
2386 C
2387 C***** Subroutine doit3d *****
2388 C
2389 C this routine has DISPLA calls
2390 C
2391 C this routine puts the data for the 3-d plot in a variably dimensioned
2392 C 2-d array to match the requirements of surmat()
2393 C
2394      subroutine doit3d(bxmin, bxpts, lymin, lypts, angl, wlenl, m, idx, fwrk)
2395      integer maxplt, maxcur, maxang, maxwin, maxsur, maxdat
2396      parameter (maxplt=4, maxcur=7, maxang=25)
2397      parameter (maxwin=71, maxsur=10, maxdat=20)
2398      integer bxmin, bxmax, lymin, lymin, bxpts, lypts, m, idx
2399      integer iwork(2*(maxwin*maxang)+4)
2400      real angl(maxsur, maxang), wlenl(maxsur, maxwin)
2401      real fct(maxdat, maxwin, maxang), fwrk(bxpts, lypts)
2402      real x33mat, y33mat, setx3, sety3, setup
2403      external x33mat, y33mat
2404      common/one/fct
2405 C
2406      bxmax = bxmin + bxpts - 1
2407      lymin = lymin + lypts - 1
2408      lypts = lymin + lymin + 1
2409 c      setup = setx3(m, wlenl, bxmin)
2410 c      setup = sety3(m, angl, lymin)
2411 C
2412 c copy data from storage into smaller array
2413      do 4600 j = bxmin, bxmax
2414          do 4600 k = lymin, lymin
2415              fwrk(j-bxmin+1, k-lymin+1) = fct(idx, j, k)
2416      4600 continue
2417 C
2418 c do plot
2419      call surmat(fwrk, 1, bxpts, 1, lypts, iwork)
2420      return
2421      end
2422 C
2423 C
2424 C***** Functions x33mat, setx *****
2425 C
2426 C This routine has NO displa calls
2427 C
2428 C This routine returns the wavelength for a given x index to contour plotting

```

## Appendix V

```

2429 C routine. This function was to be x3dmat and be used with surtm() but
2430 C that subroutine does not seem to work in displa on the cray2.
2431 C
2432     function x33mat()
2433     integer maxsur,maxwin,m,j
2434     parameter(maxsur=10,maxwin=71)
2435     integer bxmin,dxmin
2436     Real wlen1(maxwin),dwlen1(maxsur,maxwin)
2437     save
2438 C
2439 c simple isn't it
2440     x33mat=wlen1(j)+bxmin-1)
2441     return
2442 C
2443 C
2444 c this entry copies in the list of wavelengths for the given surface asgn #
2445     entry setx3(m,dwlen1,dxmin)
2446     bxmin=dxmin
2447     do 4700 i=1,maxwin
2448         wlen1(i)=dwlen1(m,i)
2449 4700 continue
2450     setx3=1.
2451     return
2452     end
2453 C
2454 C
2455 C----- Function y33mat, sety -----
2456 C
2457 C This function has NO displa calls
2458 C
2459 C this routine is just like x33mat except that it deals with the inc angles
2460 C
2461     function y33mat()
2462     integer maxsur,maxang,m,j
2463     parameter(maxsur=10,maxang=25)
2464     integer diymin,tymin
2465     Real ang1(maxang),dang1(maxsur,maxang)
2466     save
2467 C
2468     y33mat=ang1(j)+tymin-1)
2469     return
2470 C
2471 C
2472     entry sety3(m,dang1,diymin)
2473     tymin=diymin
2474     do 4800 i=1,maxang
2475         ang1(i)=dang1(m,i)
2476 4800 continue
2477     sety3=1.
2478     return
2479     end
2480 C
2481 C
2482 C----- Subroutine rng3d -----

```

## Appendix V

```

2483 C
2484 C This routine has NO displa calls
2485 C
2486 C this is a normal min max search process in the desired range for the plot
2487 C
2488     subroutine mg3d(lbx,bxmin,bxmax,lymin,lymax,zmin,zmax)
2489     integer maxpit,maxcur,maxang,maxwin,maxsur,maxdat
2490     Parameter (maxpit=4,maxcur=7,maxang=25)
2491     Parameter (maxwin=71,maxsur=10,maxdat=20)
2492     integer lbx,bxmin,bxmax,lymin,lymax,i,j
2493     Real zmin,zmax,fst(maxdat,maxwin,maxang)
2494 C
2495     Common/one/fst
2496 C
2497     zmin=1.e36
2498     zmax=-1.e36
2499     do 5000 i=bxmin,bxmax
2500         do 5000 j=lymin,lymax
2501             if(fst(lbx,i,j).gt.zmax) zmax=fst(lbx,i,j)
2502             if(fst(lbx,i,j).lt.zmin) zmin=fst(lbx,i,j)
2503     5000 continue
2504     return
2505     end
2506 C
2507 C
2508 C***** Subroutine head3d *****
2509 C
2510 C This routine has DISSPLA calls
2511 C
2512 C this routine prints the center heading for 3d plots and contour plots
2513 C
2514     subroutine head3d(neurf,matnm,autnm,hmaq,sigs,elt,n1,n2)
2515     integer elt,neurf
2516     Real hmaq,sigs
2517     Character*13 matnm,autnm
2518 C
2519 c setup subplot area
2520     call physort(0.,0.)
2521     call area2d(12.,10.5)
2522     call height(.26)
2523     call messag('Backscatter Mueller Matrix Element ~',100,0.9,10.)
2524     call makef(elt,n1,n2)
2525     call height(.21)
2526 c if there is only one material on a contour or for all 3-d plots, print it
2527     if(neurf.eq.1) then
2528         call messag(matnm/' '//autnm/'~',100,3.2,9.75)
2529         call height(.17)
2530         call messag('(Z1) - ~',100,3.1,9.45)
2531         call resino(hmaq,104,'ABUT','ABUT')
2532         call messag('(H.7M7)M(M1)M(EH.5)2(EXHXMX)~',100,'ABUT','ABUT')
2533         call messag('(Z2) - ~',100,'ABUT','ABUT')
2534         call resino(sigs,104,'ABUT','ABUT')
2535     else
2536 c otherwise print comparison

```

## Appendix V

```

2537      call messag("Comparison of Materials",100,3.4,9.75)
2538      endif
2539      call reset("HEIGHT")
2540      call endgr(0)
2541      return
2542      end
2543      C
2544      C
2545      C***** Subroutines prtn3d, sect3d *****
2546      C
2547      C This routine has DISPLA calls
2548      C
2549      C This routine sets up the page area for displaying the 3-d plot
2550      C
2551      subroutine prtn3d(vx,vy,vz)
2552      Real vx,vy,vz,xac,yac,xscale,yscale
2553      C
2554      c scaling of physical origin is important
2555      c setting view pt is also done here
2556      call physar(1,"xac,.25*yac)
2557      call tit3d(" ",100,11.,9.5)
2558      call volm3d(2.,1.,1.)
2559      call vusbe(vx,vy,vz)
2560      return
2561      C
2562      C
2563      c this entry passes in the scaling factor to use
2564      entry sect3d(xscale,yscale)
2565      xac=xscale
2566      yac=yscale
2567      return
2568      end
2569      C
2570      C
2571      C***** Subroutine axes33 *****
2572      C
2573      C This routine has DISPLA calls
2574      C
2575      C This routine setups for and plots the 3 axes on 3-d plots
2576      C
2577      subroutine axes33(x1,x2,y1,y2,z1,z2)
2578      Real x1,x2,y1,y2,z1,z2
2579      C
2580      call saxang(-90.)
2581      call yaxang(0.)
2582      call heigh(.21)
2583      c name them
2584      call x3name("(P4)Wavelength in (M7)M(M1)M",100)
2585      call y3name("(P4)Incident Angle in deg.",100)
2586      call z3name(" ",100)
2587      call simplx
2588      c draw them
2589      call gra3d(x1,(x2-x1)/5.,x2,y1,(y2-y1)/5.,y2,z1,(z2-z1)/5.,z2)
2590      call complx

```

# Appendix V

```

2591      return
2592      end
2593      C
2594      C
2595      C***** Subroutine lgcn *****
2596      C
2597      C This routine has DISPLA calls
2598      C
2599      C this routine prints the legend of the contour levels in the upper right corner
2600      C
2601      subroutine lgcn(lvl)
2602      Real lvl(11)
2603      C
2604      call physcn(0.,0.)
2605      call area2d(11.5,10.5)
2606      do 5900 i=1,5
2607          call linesz(i)
2608          call stripz(9.7,10.2-.2*i)
2609          call corrup(10.2,10.2-.2*i)
2610          call reset('DASH')
2611          call resino(lvl*(P2-1)-.3,10.3,10.1-.2*i)
2612      5900 continue
2613      call reset('SETCLR')
2614      call endgr(0)
2615      return
2616      end
2617      C
2618      C
2619      C***** Subroutines axescn, sgrdcn *****
2620      C
2621      C This routine has DISPLA calls
2622      C
2623      C this routine names and plots the axes for contour plots
2624      C
2625      subroutine axescn(x1,x2,y1,y2,nsurf,matnm,autnm,hmaq,sigs)
2626      Integer nsurf,ifgrd,difgrd,ngrdy,ngrdx,dngrdx,dngrdy
2627      Real x1,x2,y1,y2,hmaq,sigs
2628      Character*13 matnm,autnm
2629      C
2630      c name them
2631      call yaxang(0.)
2632      call yname('P4')Incident Angle in deg.",100)
2633      call xname('P4')Wavelength in (M7)M(M1)M",100)
2634      call slmptz
2635      c plot them
2636      call graf(x1,(x2-x1)/10.,x2,y1,(y2-y1)/5.,y2)
2637      call complx
2638      call height(.21)
2639      C
2640      c if materials are being compared write material data over each plot
2641      if(nsurf.gt.1) then
2642          call message(matnm/' '//autnm/' ",100,0.,6.6)
2643          call message('Z1) - ",100,'ABUT','ABUT')
2644          call resino(hmaq,104,'ABUT','ABUT')

```

# Appendix V

```

2645      call messag('M7)M(M1)M(EH.5)2*",100,'ABUT','ABUT')
2646      call messag('  (Z2) - ",100,'ABUT','ABUT')
2647      call realno(sigs,104,'ABUT','ABUT')
2648      call reset('HEIGHT')
2649      endif
2650      c draw grid lines if desired
2651      if (ifgrd.eq.1) then
2652          call dash
2653          call grid(ngrdb,ngrdy)
2654          call reset('DASH')
2655      endif
2656      return
2657      C
2658      C
2659      c this entry passes grid line info in
2660      entry sgrdcn(dfgrd,dngrdb,dngrdy)
2661      ifgrd=dfgrd
2662      ngrdb=dngrdb
2663      ngrdy=dngrdy
2664      return
2665      end
2666      C
2667      C
2668      C***** Subroutine dotcn *****
2669      C
2670      C This routine has NO displa calls
2671      C
2672      C This routine determines the points on each of 9 contour levels of the
2673      C corresponding 3-d plot
2674      C
2675      subroutine dotcn(txmin,dtxpts,tymin,diypoints,angl,wlenl,msurf,
2676      *      kdx,lvl,s1,s2)
2677      integer maxplt,maxcur,maxang,maxwin,maxsur,maxdat
2678      parameter(maxplt=4,maxcur=7,maxang=25)
2679      parameter(maxwin=71,maxsur=10,maxdat=20)
2680      integer txmin,txpts,dtxpts,txmax,tymin,diypoints,typts,tymax
2681      integer msurf,kdx,lvl
2682      integer kdx(4),kdi(4)
2683      integer hk(1000),hl(1000),hd(1000),n
2684      real angl(maxsur,maxang),wlenl(maxsur,maxwin)
2685      real fwrk(maxwin,maxang)
2686      real fst(maxdat,maxwin,maxang)
2687      real lvl(11),s1,s2,level
2688      real xray(1000),yray(1000)
2689      C
2690      Common/onefst
2691      Common/fivelevel,ihits,hk,hl,hd,fwrk,txpts,typts
2692      Data kdx,kdi/0,1,0,-1,1,0,-1,0/
2693      C
2694      c move proper range of data from storage into the working 2-d array
2695      txpts=dtxpts
2696      typts=diypoints
2697      txmax=txmin+txpts-1
2698      tymax=tymin+typts-1

```

# Appendix V

```

2699      setup=set3(msurf,wlenl,bxmin)
2700      setup=set3(msurf,angl,iymin)
2701      do 6100 j=bxmin,bxmax
2702      do 6100 k=iymin,iymax
2703          fwrk(j-bxmin+1,k-iymin+1)=fst(idx,j,k)
2704      6100 continue
2705      C
2706      c loop through each of 9 levels
2707      c assigning colors or dashed lines to each
2708      do 6110 ilvl= 9,1,-1
2709          ihits=0
2710          level=lv(ilvl)
2711          if(ilvl.eq.2*(ilvl/2)) then
2712              kcolor=10
2713          else
2714              kcolor=ilvl/2+1
2715          endif
2716      C
2717      c find all places where the contour line pass between adjacent wavelengths
2718      do 6112 k=1,brpts-1
2719      do 6112 l=1,lypts
2720          call hitme(k,l,1)
2721          call hitme(k+1,l,3)
2722      6112 continue
2723      C
2724      c find all place where the contour line pass between adjacent inc angles
2725      do 6114 l=1,lypts-1
2726      do 6114 k=1,brpts
2727          call hitme(k,l,0)
2728          call hitme(k,l+1,2)
2729      6114 continue
2730      C
2731      C
2732      c iflag1=0 ==> working on edge to edge contours
2733      c iflag1=1 ==> working on closed contours
2734      iflag1=0
2735      ksv=0
2736      6130 if(ihits.eq.0) goto 6110
2737      ksv=-1
2738      lpts=0
2739      if(iflag1.eq.1) goto 6133
2740      c find an edge entry point
2741      n=1
2742      6131 if(hd(n).eq.0.and.hk(n).eq.1) goto 6135
2743      if(hd(n).eq.1.and.hk(n).eq.lypts) goto 6135
2744      if(hd(n).eq.2.and.hk(n).eq.brpts) goto 6135
2745      if(hd(n).eq.3.and.hk(n).eq.1) goto 6135
2746      if(n.eq.ihits) goto 6133
2747      n=n+1
2748      goto 6131
2749      C
2750      c pick a hit as starting for close contour, save it and copy it
2751      6133 iflag1=1
2752      ksv=hk(1)

```

# Appendix V

```

2753     lev=hl(1)
2754     ldev=hd(1)
2755     ihits=ihits+1
2756     hl(ihits)=lev
2757     hl(ihits)=lev
2758     hd(ihits)=ldev
2759     n=1
2760 C
2761 c save interpolated x and y values in the plotting array
2762 6135 id=hd(n)
2763     k=hl(n)
2764     l=hl(n)
2765     kp=k+idl(id+1)
2766     lp=l+idl(id+1)
2767     lpts=lpts+1
2768     x1=x33mat(k)
2769     y1=y33mat(l)
2770     delf=(level-fwrk(k,l))/(fwrk(k+idl(id+1),l+idl(id+1))-fwrk(k,l))
2771     xray(lpts)=(x33mat(k+idl(id+1))-x1)*delf+x1
2772     yray(lpts)=(y33mat(l+idl(id+1))-y1)*delf+y1
2773 c remove hit from list
2774     hd(n)=hd(ihits)
2775     hl(n)=hl(ihits)
2776     hl(n)=hl(ihits)
2777     ihits=ihits-1
2778 c see if you are at the end of edge to edge
2779     if(flag1.eq.0) then
2780         if((ld.eq.0.and.k.eq.kpts).or.(ld.eq.2.and.k.eq.1).or.
2781             * (ld.eq.1.and.l.eq.1).or.(ld.eq.3.and.l.eq.lypts)) then
2782             call curvcn(xray,yray,icolor,lpts)
2783             goto 6130
2784         endif
2785 c see if you are at the end of a closed contour
2786         else
2787             if(k.eq.kav.and.l.eq.lav.and.id.eq.ldev) then
2788                 if(lpts.gt.1) then
2789                     call curvcn(xray,yray,icolor,lpts)
2790                     goto 6130
2791                 endif
2792             endif
2793         endif
2794 C
2795 c determine what the next hit should be
2796 c this is the meat of the code but it is not really possible to document here
2797 c the best way to understand it is to do a pencil and paper run
2798     idp=incd(id)
2799     kpp=k+idl(idp+1)
2800     lpp=l+idl(idp+1)
2801     if(fwrk(kpp,lpp).gt.level) then
2802         knew=kpp
2803         lnew=lpp
2804         idnew=incd(incd(idp))
2805     else
2806         kpp=k+idl(idp+1)

```

# Appendix V

```

2807      lppp=l+idl(idp+1)
2808      if(fwrk(lppp,lppp).gt.level) then
2809          knew=lppp
2810          lnew=lppp
2811          idnew=id
2812      else
2813          knew=k
2814          lnew=l
2815          idnew=idp
2816      endif
2817  endif
2818  C
2819  c find the new hit in hit list
2820      n=1
2821      6138 if(hk(n).eq.knew.and.hl(n).eq.lnew.and.hd(n).eq.idnew) goto 6135
2822      if(n.eq.ihits) then
2823          goto 6110
2824      endif
2825      n=n+1
2826      goto 6138
2827  6110 continue
2828      return
2829  end
2830  C
2831  C
2832  C----- Subroutine hitme -----
2833  C
2834  C This routine has NO displa calls
2835  c
2836  c this routine determines if a base pt and a direction constitutes a hit for
2837  c the given contour level
2838  c
2839      subroutine hitme(k,l,m)
2840      integer maxwin,maxang
2841      Parameter(maxwin=71,maxang=25)
2842      integer k,l,m,hk(1000),hl(1000),hd(1000),ixpts,iypts
2843      integer idk(4),idl(4)
2844      Real level,fwrk(maxwin,maxang)
2845  C
2846      Common/fixa/level,ihits,hk,hl,hd,fwrk,ixpts,iypts
2847      Data idk,idl/0,1,0,-1,1,0,-1,0/
2848  C
2849      if(fwrk(k,l).ge.level) then
2850          kp=k+idl(m+1)
2851          lp=l+idl(m+1)
2852      if(fwrk(kp,lp).lt.level) then
2853          ihits=ihits+1
2854          hk(ihits)=k
2855          hl(ihits)=l
2856          hd(ihits)=m
2857      endif
2858  endif
2859      return
2860  end

```

## Appendix V

```

2861 C
2862 C
2863 C***** Function incd *****
2864 C
2865 C This routine is a mod 4 function
2866 C
2867     function incd(idd)
2868     integer idd
2869     incd = idd + 1 - 4*((idd+1)/4)
2870     return
2871 end
2872 C
2873 C
2874 C***** Subroutine curven *****
2875 C
2876 C This routine has DISPLA calls
2877 C
2878 C This routines plots the contours as any other 2-d curve
2879 C a special color handling section makes every other contour a dashed line
2880 C
2881     subroutine curven(xray,yray,icolor,ipis)
2882     integer icolor,ipis
2883     Real xray(ipis),yray(ipis)
2884     if(icolor.eq.10) then
2885         call dash
2886     else
2887         call lines(icolor)
2888     endif
2889     call curve(xray,yray,ipis,0)
2890     call reset("SETCLR")
2891     call reset("DASH")
2892     return
2893 end

```

Blank

## APPENDIX VI: DETECT/DECIDE2 USER GUIDES AND SOURCE CODES.

### AVI.1.1 Introduction to DETECT.

Program DETECT, written by Mark Haugland, analyzes backscatter Mueller matrices calculated by RETRO. Both DETECT and RETRO are written in FORTRAN 77 and set up to run on a Cray 2 under the UNICOS operating system. The program serves two purposes: (1) it locates optimum angles of incidence and wavelengths for use in discriminating between a contaminated and a dry surface; (2) it then identifies those Mueller matrix elements that can be used to discriminate between contaminated and dry surfaces at optimum beams angles of incidence and beam wavelengths.

#### AVI.1.1.1 Outline of DETECT's functions.

DETECT functions in the following manner:

1. Read in two output files from RETRO.
  - a) Either normalize all matrix elements to  $f_{11}$  or use the selected auto-correlation function.
2. Write a heading block to the output file (unit 9).
3. Compute Equations (37) and (39). The program does not enforce the condition that these values must be larger than  $|x|_{\max}$  and  $|y|_{\max}$ . The user must decide what these values are and make judgements accordingly.
  - a) Sort the terms in Equations (37) and (39) so that they are in decreasing order of magnitude.
4. Write to unit 9.
  - a) The subscripts of non-zero terms in Equation (39) in the order of largest to smallest terms. The subscript of the largest term is followed by the subscript of the next largest term and so on,
  - b) the subscripts of all non-zero terms in (37) in the order of largest to smallest term. The subscript of the largest term is followed by the subscript of the next largest term and so on,
  - c) Equation (37), largest term, subscript of largest term,
  - d) Equation (39), largest term, subscript of largest term,
  - e) the angle and wavelength at which (37) took on its maximum value over the range of wavelengths and angles in the RETRO output files and the maximum value,
  - f) the angle and wavelength at which (39) took on its maximum value over the range of wavelengths and angles in the RETRO output files and the maximum value,
  - g) the largest value assumed by any term in (37), the subscript of that term, the angle and wavelength at which that value occurred,
  - h) the largest value assumed by any term in (39), the subscript of that term, the angle and wavelength at which that value occurred.
5. Find all local maxima and minima in  $k$  (imaginary refractive index) for both materials,
6. write all local maxima and minima in  $k$  for each material to unit 9.

#### AVI.1.1.2 Input Files.

DETECT needs five input files. The first file is 'FILENAMES,' which is used in every run and contains names of the four other files required by DETECT. In addition to names of the other input files, 'FILENAMES' contains code numbers that determine which auto-correlation function the program accesses for the background material, which auto-correlation function the program uses for the target material, or whether to normalize all Mueller matrix elements to  $f_{11}$

#### AVI.1.1.2.1 FILENAMES.

An example of a correctly edited file is given at the end of this section. The line-by-line format is as follows:

- A- filename containing RETRO output for the background (base) material (up to 15 characters);
- B- filename containing RETRO output for the target material (up to 15 characters);
- C- filename for data output, unit 9 (15 characters max);
- D- filename for commentary output, unit 8 (15 characters max);
- E- filename containing the background material's index of refraction as a function of wavelength (15 characters or less);
- F- filename containing the target material's index of refraction as a function of wavelength (15 characters or less);
- G- The code number (IBCOR) determines whether the auto-correlation function for the background material is Gaussian (IBCOR=1), N=8 (IBCOR=2), or N=6 (IBCOR=3) (integer);
- H- The code number ITCOR determines whether the auto-correlation function for the target material is Gaussian (ITCOR=1), N=8 (ITCOR=2), or N=6 (ITCOR=3) (integer);
- I- The code number NORM equals zero for no normalization and 1 for normalizing all Mueller matrix elements to  $f_{11}$ . That is, a value of 1 on this line causes the program to analyze  $\frac{f_{ij}}{f_{11}}$  rather than  $f_{ij}$  (integer).

#### AVI.1.1.2.2 Accessing RETRO output files.

Program DETECT analyzes RETRO's output data as a function of beam wavelength and incident angle. The wavelengths and incident angles in RETRO's output file for background and target materials must be identical. If not the same, DETECT will quit and write an error message to unit 8. For more information regarding files generated by RETRO, see Appendix V.

#### AVI.1.1.2.3 Index of refraction files.

These files are the same files RETRO accesses to compute the dielectric constant for each material.

#### AVI.1.1.3 Output files.

DETECT creates two output files. The first file is for commentary output should something go wrong, the second is the DETECT data file. The names for these files are the inputs to program DETECT found on lines C and D of 'FILENAMES.'

#### AVI.1.1.3.1 Commentary output file.

This file is opened as unit 8. Unit 8 is sent various values for debugging the program if something should go wrong during the computer run. With normal operations the only written input to unit 8 are error messages indicating if wavelengths and angles of incidence in each RETRO output file are not equal or if the number on line I of 'FILENAMES' is not 0 or 1. In DETECT, several write statements to unit 8 have been commented out. They may be reinstated by removing "C" in the leftmost column on the line where the write statement appears.

#### AVI.1.1.3.2 Data output file.

This file (written to unit 9) contains four distinct sections. The header block appears at the beginning of the file. The second section contains data from the analysis of RETRO's output. The third section contains maximum quantities contained in the second section. The final section contains local maxima and minima of k-values for both materials. The format of the heading block is as follows.

- A- A description of the base material,
- B- and the target material.
- C- Present if line I in 'FILENAMES' is 1. This line is a comment to remind the user that RETRO output was normalized to  $f_{11}$ .
- D- Present if line I in 'FILENAMES' is 0. This line has the same value as that of line G of 'FILENAMES,' to remind the user which auto-correlation function for the background material DETECT used in its analysis.
- E- This line is the same as that of line D, for the target material.
- F- The number of  $\langle h^2 \rangle$ 's,  $\sigma_s^2$ 's,  $\lambda$ 's, and  $\theta_0$ 's used by RETRO. If the number of  $\langle h^2 \rangle$ 's or  $\sigma_s^2$ 's is greater than 1, the second and third sections of this file are repeated in a nested loop where  $\sigma_s^2$  varies more rapidly than  $\langle h^2 \rangle$ .
- G- A comment that the next line is the  $\langle h^2 \rangle$ 's for the background material.
- H-  $\langle h^2 \rangle$ 's for the background material.
- I- A comment that the next line is the  $\sigma_s^2$ 's for the background material.
- J-  $\sigma_s^2$ 's for the background material.
- K-
- L-
- M-
- N- Lines K, L, M, and N are the same as lines G, H, I, and J, for the target material.
- O- A comment that the next line(s) list the wavelengths,
- P- and those wavelengths used.
- Q- A comment that the next line(s) list the incident angles,
- R- and those incident angles used.

The second section of the data output file is program DETECT's output of the Mueller matrices. These data are generated in a nested loop where  $\theta_0$  varies more rapidly than  $\lambda$ . In this section, there are three lines of data for each pair of Mueller matrices read in. In addition, every time the wavelength is incremented, a line 'WAVELENGTH string' is written with wavelength value as the string in  $\mu m$ . The format of this section is as follows:

- A2- Wavelength is written here each time it changes.
- B2- The subscripts of all non-zero components of  $\bar{y}$  Equation (39) are listed in decreasing order of the magnitude of the component. As an example, suppose  $y_5 > y_1 > y_4$  and  $y_2 = y_3 = y_6 = 0$ . For the  $\bar{y}$  described above, this line would read 5 1 4.
- C2- This line is the same as B2, except it is associated with  $\bar{x}$  of Equation (37).
- D2- Values:  $|\bar{x}|$ ,  $|\bar{y}|$ ,  $j$ ,  $(x_j)_{\max}$ ,  $i$ , and  $(y_i)_{\max}$   
 where;  
 $|\bar{x}|$  is calculated from Equation (37),  
 $|\bar{y}|$  is calculated from Equation (39),  
 $(x_j)_{\max}$  is the largest component of  $\bar{x}$ ,  
 and  $(y_i)_{\max}$  is the largest component of  $\bar{y}$ .

The third section in the data output file is repeated once for each combination of  $\langle h^2 \rangle$  and  $\sigma_p^2$  read in. This section gives the user a general idea where to find the most interesting information in the second section. There are four lines of data in this section.

- A3- Values  $\theta_0$ ,  $\lambda$ ,  $(x_k)_{\max}$ ,  $k$ .  
 $(x_k)_{\max}$  is the largest value any component of  $\bar{x}$  assumed over all combinations of the incident angles and wavelengths examined.  
 $\theta_0$  and  $\lambda$  are the incident angle and wavelength at which  $(x_k)_{\max}$  occurred.
- B3- Values  $\theta_0$ ,  $\lambda$ ,  $(y_k)_{\max}$ ,  $k$ .  
 Same as A3 except for  $\bar{y}$ .
- C3- Values  $\theta_0$ ,  $\lambda$ ,  $|\bar{x}|_{\max}$ .  
 $\theta_0$  and  $\lambda$  are the angle of incidence and wavelength values for which  $|\bar{x}|$  was maximum over all combinations of the incident angles and wavelengths examined.
- D3- Values  $\theta_0$ ,  $\lambda$ ,  $|\bar{y}|_{\max}$ .  
 Same as C3 except for  $\bar{y}$ .

The fourth and final section of this file lists information regarding the indices of refraction for the background and target materials. The purpose of this section is to identify the correlation between resonant wavelengths and interesting behavior in each material's Mueller matrices. Four subsections are ordered as follows:

- A4- Wavelengths and values for local maxima of  $k$  for the base (background) material.
- B4- Wavelengths and values for local minima of  $k$  for the base material.
- C4- Wavelengths and values for local maxima of  $k$  for the target material.
- D4- Wavelengths and values for local minima of  $k$  for the target material.

#### AVI.1.2 Instructions for running the program.

## Appendix VI

The steps required to run DETECT are now presented. DETECT can be executed after input file 'FILENAMES' has been properly edited and all of the other input files are loaded. Execution time is short, thus there is no need to run DETECT in the batch mode. Typing "detect.run" after the UNICOS prompt "bob>" will compile, link, and execute the program accessing data stored in 'FILENAMES.' The executable code is stored in "detect.xqt." To run the program again during the same login after changing 'FILENAMES,' simply type "detect.xqt." Do this rather than typing "detect.run" to avoid recompilation and linking. Before logging off, remove 'detect.xqt,' for it is a very large file. The object file 'detect.o' was removed by 'detect.run' immediately following linkage.

## Appendix VI

### AVI.1.3 DETECT SOURCE CODE.

```
1  C This work was done for the CRDEC on the Aberdeen Proving Grounds
2  C Edgewood Area.
3  C This work was done by S. Mark Haugland under contract DAAD0589P0427.
4  C
5  C The purpose of this program is to aid the user in locating combinations
6  C of backscatter angle and wavelength of optimal use in discriminating
7  C between a background (base) surface and the surface with
8  C an optically thick layer of contaminant on top of it (target material).
9  C The term optically thick restricts the
10 C Mueller matrix elements which are of any use to those that are
11 C larger in magnitude for the contaminant than for the background.
12 C
13 C DETECT functions in the following manner:
14 C 1. Open the file FILENAMES and read the following input variables in:
15 C   IN1NM- the name of RETRO output file for the background material
16 C   IN2NM- the name of RETRO output file for the target material
17 C   OUTNM- the name of the file this program writes results to.
18 C   COMNM- the name of the file this program writes commentary
19 C   output to
20 C   BASENK- the name of the file containing the background material's
21 C     index of refraction as a function of wavelength
22 C   TARGNK- the name of the file containing the target material's
23 C     index of refraction as a function of wavelength
24 C   IBCOR- code number that selects the auto-correlation function
25 C     to use in constructing the Mueller matrices for the
26 C     background material
27 C   ITCOR- code number that selects the auto-correlation function
28 C     to use in constructing the Mueller matrices for the
29 C     target material
30 C   NORM- code number to determine whether DETECT will process
31 C     Mueller matrices or Mueller matrices normalized to
32 C     P11.
33 C   2. Open and read 2 output files from RETRO. These files contain
34 C     information needed to construct a set of Mueller matrices
35 C     for the target and the background materials.
36 C     a) either normalize to P11 or construct unnormalized Mueller
37 C     matrices using the auto-correlation functions requested
38 C     by IBCOR and ITCOR.
39 C 3. Write a heading block to the data output file (unit 9)
40 C 4. Compute data discussed in the DETECT user's manual.
41 C 5. Write data to unit 9.
42 C 6. Open the files containing the index of refraction as a
43 C     function of wavelength for the base material and find
44 C     all local maxima and minima of k in that file. Write
45 C     these values and the corresponding wavelengths to unit 9.
46 C 7. Open the files containing the index of refraction as a
47 C     function of wavelength for the target material and find
48 C     all local maxima and minima of k in that file. Write
49 C     these values and the corresponding wavelengths to unit 9.
50 C
51  PROGRAM DETECT
```

# Appendix VI

```

52     REAL HMSQLT(15),HMSQLB(15),WLENB(100),WLENT(100),ANGLB(100)
53     REAL ANGLT(100),SIGSLB(15),SIGSLT(15),PB(6,100,100)
54     REAL FT(6,100,100),PB1(6),PT1(6),QB(3),QT(3),DIFF(6),RATLOG(6)
55     REAL ANIMIN(40),ANIMAX(40),WLNMIN(40),WLNMX(80)
56     REAL WLNMIN(80),ANGLMIN(80),ANGLMX(80),FMAX(80),FMIN(80),ARR(6)
57     INTEGER IBCOR,ITCOR,NORM,NHMSQB,NSICSB,NWLENB,NANCB,NHMSQT
58     INTEGER NSICST,NWLENT,NANGT,NMIN,NMAX,KKA(6)
59     CHARACTER*15 IN1NM,IN2NM,OUTNM,COMNM,BASENK,TARGNK,PNAME
60     CHARACTER*13 BASEMAT,TARGMAT
61     C
62     C This block reads in inputs from FILENAMES and opens all of them except
63     C the files containing the indices of refraction for the target and
64     C background materials which are opened in the subroutine SORTNK.
65     C
66     OPEN(UNIT=7,FILE='FILENAMES')
67     READ(7,*)IN1NM
68     OPEN(UNIT=10,FILE=IN1NM)
69     READ(7,*)IN2NM
70     OPEN(UNIT=11,FILE=IN2NM)
71     READ(7,*)OUTNM
72     OPEN(UNIT=9,FILE=OUTNM)
73     READ(7,*)COMNM
74     OPEN(UNIT=8,FILE=COMNM)
75     READ(7,*)BASENK
76     READ(7,*)TARGNK
77     C
78     C This block reads in control variables from unit 7 which determine
79     C which auto-correlation function to use for the base material and which
80     C auto-correlation function to use for the target material. IBCOR determines
81     C which auto-corr function to use for the base. IBCOR=1,2,3 for Gaussian,N=8
82     C ,and N=6 respectively. ITCOR performs the same function as IBCOR except for
83     C the target material. There are two options for normalizing the Mueller matrix
84     C elements. NORM=0 no normalization is used, NORM=1 all Mueller matrix elements
85     C are normalized to P11.
86     C
87     READ(7,*)IBCOR
88     READ(7,*)ITCOR
89     READ(7,*)NORM
90     C
91     C This block reads the heading block from units 10 and 11. Units 10 and 11 are
92     C the output from two separate runs of RETRO. Unit 10 contains the data on
92     C the base material and unit 11 contains the data on the target material.
93     C
94     READ(10,*)BASEMAT
95     READ(11,*)TARGMAT
96     READ(10,1000)NHMSQB,NSICSB,NWLENB,NANCB
97     READ(11,1000)NHMSQT,NSICST,NWLENT,NANGT
98     1000 FORMAT(4H10.4)
99     READ(10,1010)(HMSQLB(IL),IL=1,NHMSQB)
100    READ(10,1010)(SIGSLB(IL),IL=1,NSICSB)
101    READ(10,1010)(WLENB(IL),IL=1,NWLENB)
102    READ(10,1010)(ANGLB(IL),IL=1,NANCB)
103    READ(11,1010)(HMSQLT(IL),IL=1,NHMSQT)
104    READ(11,1010)(SIGSLT(IL),IL=1,NSICST)

```

# Appendix VI

```

105 READ(11,1010)(WLENT(IL),IL=1,NWLENT)
106 READ(11,1010)(ANGLT(IL),IL=1,NANGT)
107 1010 FORMAT(5E12.4)
108 C
109 C This block checks if the wavelengths and the angles of incidence for the
110 C base and the target materials are the same. If the wavelengths and the angles
111 C are not the same for both sets of data, the program stops and writes an error
112 C message to unit 6.
113 C
114 IF(NANGB.NE.NANGT)GO TO 990
115 IF(NWLENT.NE.NWLENB)GO TO 990
116 DO 77 IL=1,NWLENB
117 D=WLENB(IL)-WLENT(IL)
118 IF(ABS(D).GE..001)GO TO 990
119 77 CONTINUE
120 DO 78 IL=1,NANGB
121 D=ANGLB(IL)-ANGLT(IL)
122 IF(ABS(D).GE..001)GO TO 990
123 78 CONTINUE
124 C
125 C This section puts a heading in the output file so that the user can index
126 C the data.
127 C
128 WRITE(9,9000)BASEMAT
129 9000 FORMAT('BASE MATERIAL',A13)
130 WRITE(9,9001)TARGMAT
131 9001 FORMAT('TARGET MATERIAL',A13)
132 IF(NORM.EQ.1)THEN
133 WRITE(9,9010)
134 9010 FORMAT('ALL MATRIX ELEMENTS ANALYZED ARE NORMALIZED TO F11')
135 ELSE IF(NORM.EQ.0)THEN
136 WRITE(9,9011)BCOR
137 9011 FORMAT('CODE FOR AUTO-CORR. FUNCTION FOR THE BACKGROUND
138 * MATERIAL',I5)
139 WRITE(9,9012)TCOR
140 9012 FORMAT('CODE FOR AUTO-CORR. FUNCTION FOR THE TARGET
141 * MATERIAL',I5)
142 ELSE
143 WRITE(8,8222)
144 8222 FORMAT('ILLEGAL NORM')
145 GO TO 992
146 ENDIF
147 WRITE(9,1000)NHMSQB,NSIGSB,NWLENB,NANGB
148 WRITE(9,'<h**2> for the base material')
149 WRITE(9,1010)(HMSQLB(IL),IL=1,NHMSQB)
150 WRITE(9,'mean square slopes for the base material')
151 WRITE(9,1010)(SIGSLB(IL),IL=1,NSIGSB)
152 WRITE(9,'<h**2> for the target material')
153 WRITE(9,1010)(HMSQLT(IL),IL=1,NHMSQB)
154 WRITE(9,'mean square slopes for the target material')
155 WRITE(9,1010)(SIGSLT(IL),IL=1,NSIGSB)
156 WRITE(9,'wavelengths used')
157 WRITE(9,1010)(WLENB(IL),IL=1,NWLENB)
158 WRITE(9,'incident angles')

```

## Appendix VI

```

159      WRITE(9,1010)(ANGLB(IL),IL-1,NANGB)
160      C
161      C This block reads in the actual data needed to construct the Mueller matrices
162      C for both the base and target materials. Each matrix element is tested for
163      C angle and wavelength responses. The mean square height and slope are held
164      C constant for all tests. The meaning of the subscripts for the arrays
165      C FB(K,I,I) and FT(K,I,I) are as follows: K is the matrix element
166      C (K-1-->F11, K-2-->F12, K-3-->F22, K-4-->F33, K-5-->F34, K-6-->F44.
167      C I relates to the wavelength and I relates to the incident angle.
168      C The data is read in for each mean square height and slope as a function of
169      C angle and wavelength. The tests are performed on the data for each wavelength
170      C and angle.
171      C
172      DO 90 I1-1,NHMSQB
173      DO 91 I2-1,NSIGSB
174      T2STMX=0.
175      DIFFMX=0.
176      RADMX=0.
177      DO 92 I3-1,NWLENB
178      WRITE(9,9100)WLENB(I3)
179      9100 FORMAT('WAVELENGTH',E12.4)
180      DO 93 I4-1,NANGB
181      2000 FORMAT(6E12.4)
182      READ(10,2000)(QB(IL),IL-1,3)
183      READ(10,2000)(FB1(IL),IL-1,6)
184      READ(11,2000)(QT(IL),IL-1,3)
185      READ(11,2000)(FT1(IL),IL-1,6)
186      C
187      C This section computes and normalizes the Mueller matrix elements
188      C and does some analyses.
189      C
190      DO 94 K-1,6
191      IF(NORM.EQ.0)THEN
192      FB(K,I,I)=QB(TBCOR)*FB1(K)
193      FT(K,I,I)=QT(TTCOR)*FT1(K)
194      ELSE
195      FB(K,I,I)=FB1(K)/FB1(1)
196      FT(K,I,I)=FT1(K)/FT1(1)
197      ENDF
198      C
199      C Careful with putting a zero in the denominator. Should one of the
200      C matrix elements be zero, assign a value of zero to the term that it
201      C belongs in and proceed as usual. Also, set all terms which have
202      C FB>FT equal to zero because the target is assumed optically thick.
203      C
204      IF(ABS(FT(K,I,I)).GT.ABS(FB(K,I,I)))THEN
205      DIFF(K)=FT(K,I,I)-FB(K,I,I)
206      IF(FB(K,I,I).NE.0..AND.FT(K,I,I).NE.0)THEN
207      RAT=SQRT(ABS(FT(K,I,I)*FB(K,I,I)))
208      RATLOG(K)=DIFF(K)/RAT
209      ELSE
210      RATLOG(K)=0.
211      ENDF
212      ELSE

```

# Appendix VI

```

213     DIFF(K)-0.
214     RATLOG(K)-0.
215     ENDIF
216     C WRITE(8,8001)ABS(PB(K,I,I)),ABS(PT(K,I,I)),DIFF(K)
217     C IF(I3.EQ.NWLENB.AND.I4.EQ.NANGB)WRITE(8,8001)DIFF(K),RATLOG(K)
218     8001 FORMAT(3E12.4)
219     94 CONTINUE
220     C
221     C This block executes all operations that are done on the mtr. elements
222     C holding the angle and wavelength constant. This type of test should
223     C give one an idea which wavelength and angle of incidence to use, and
224     C which Mueller matrix elements are of most use.
225     C
226     CALL AMTXELE(RATLOG,ARR,KKA,NOP)
227     WRITE(9,9019)(KKA(MM),MM-1,NOP)
228     C WRITE(8,9019)KKA
229     C WRITE(8,8777)RATLOG
230     CALL AMTXELE(DIFF,ARR,KKA,NOP)
231     WRITE(9,9019)(KKA(MM),MM-1,NOP)
232     8777 FORMAT(6E12.4)
233     C WRITE(8,9019)KKA
234     C WRITE(8,8777)DIFF
235     9019 FORMAT(6D)
236     CALL T2STAT(DIFF,RATLOG,LMAX,DIFFMAX,RADMAX,LRMAX,RADIUS,RLOG)
237     WRITE(9,9022)RADIUS,RLOG,LMAX,DIFFMAX,LRMAX,RADMAX
238     9022 FORMAT(2E12.4,I10.4,E12.4,I10.4,E12.4)
239     C This if block finds the maximum difference between any two matrix elements
240     C and saves the wavelength and incident angle at which the maximum difference
241     C occurred. The matrix element subscript is also saved.
242     IF(ABS(DIFFMAX).GT.ABS(DIFFMX))THEN
243         DIFFMX=DIFFMAX
244         LMX=LMAX
245         I3MX=I3
246         I4MX=I4
247     ENDIF
248     C This if block finds the largest term in the array RETAOG that occurred.
249     C the wavelength, incident angle, and subscript are also saved.
250     IF(ABS(RADMAX).GT.ABS(RADMX))THEN
251         RADMX=RADMAX
252         LRMX=LRMAX
253         I3RMX=I3
254         I4RMX=I4
255     ENDIF
256     C This if block finds the incident angle and wavelength at which the Euclidean
257     C distance is at a maximum between the set of data for the target material and
258     C the base material.
259     IF(RADIUS.GT.T2STMX)THEN
260         T2STMX=RADIUS
261         I3RDMX=I3
262         I4RDMX=I4
263     ENDIF
264     C This block finds the incident angle and wavelength at which
265     C RLOG took on its maximum value.
266     IF(RLOG.GT.RLOGMX)THEN

```

## Appendix VI

```

267   RLOGMX-RLOG
268   BRLMX-B
269   WRLMX-W
270   ENDIF
271   93 CONTINUE
272   92 CONTINUE
273   WRITE(9,9101)ANGLB(4IMX),WLENB(3MX),DIFFMX,LMX
274   WRITE(9,9101)ANGLB(4RDMX),WLENB(3RDMX),RADMX,LRMX
275   WRITE(9,9101)ANGLB(4RDMX),WLENB(3RDMX),T2STMX
276   WRITE(9,9101)ANGLB(4RDMX),WLENB(3RDMX),RLOGMX
277   9101 FORMAT(3E12.4,110.4)
278   91 CONTINUE
279   90 CONTINUE
280   C
281   C This section finds all local maxima and minima in the imaginary part of
282   C index of refraction of both materials being compared. The purpose for this
283   C is to identify resonant wavelengths and to investigate the relationship
284   C between the index of refraction and the dependence of the Mueller matrix
285   C on wavelength.
286   C
287   PNAME-BASENK
288   CALL SORTNK(PNAME,NMIN,NMAX,WLNMX,WLNMIN,ANKMIN,ANKMAX)
289   WRITE(9,9110)
290   9110 FORMAT('Wavelengths and values of maxima in k for the base
291   *material')
292   DO 170 II=1,NMAX
293   WRITE(9,9111)WLNMX(II),ANKMAX(II)
294   170 CONTINUE
295   9111 FORMAT(2E12.4)
296   WRITE(9,9112)
297   9112 FORMAT('Wavelengths and values of minima in k for the base
298   *material')
299   DO 171 II=1,NMIN
300   WRITE(9,9111)WLNMIN(II),ANKMIN(II)
301   171 CONTINUE
302   PNAME-TARGETK
303   CALL SORTNK(PNAME,NMIN,NMAX,WLNMX,WLNMIN,ANKMIN,ANKMAX)
304   WRITE(9,9113)
305   9113 FORMAT('Wavelengths and values of maxima in k for the target
306   *material')
307   DO 172 II=1,NMAX
308   172 WRITE(9,9111)WLNMX(II),ANKMAX(II)
309   WRITE(9,9114)
310   9114 FORMAT('Wavelengths and values of minima in k for the target
311   *material')
312   DO 173 II=1,NMIN
313   173 WRITE(9,9111)WLNMIN(II),ANKMIN(II)
314   GO TO 991
315   990 WRITE(8,8000)
316   8000 FORMAT('INCORRECT INPUT DATA CHECK UNITS 10 AND 11 AND MAKE/
317   1'SURE THAT THE BASE AND TARGET ARE BEING COMPARED FOR THE/
318   2'SAME WAVELENGTHS AND INCIDENT ANGLES')
319   992 CLOSE(7)
320   CLOSE(8)

```

# Appendix VI

```

321     CLOSE(9)
322     CLOSE(10)
323     CLOSE(11)
324     991 STOP
325     END
326     C
327     C*****SUBROUTINE MTXLE*****
328     C
329     C This routine sorts the arrays containing the terms in the discrimination
330     C tests of T2STAT in order of largest to smallest. The purpose of this is
331     C to find out which matrix elements are most useful at a particular
332     C angle of incidence and wavelength.
333     C
334     SUBROUTINE AMTXLE(X,Y,K,N)
335     INTEGER K(6),KK(6)
336     REAL X(6),Y(6)
337     DATA KK/1,2,3,4,5,6/
338     Y-X
339     K-KK
340     C WRITE(8,8888)Y
341     C WRITE(8,8888)X
342     8888 FORMAT(6E12.4)
343     DO 100 M=1,5
344     DO 100 L=1,6-M
345     IF(ABS(Y(L)).LE.ABS(Y(L+1)))THEN
346     T=Y(L)
347     Y(L)=Y(L+1)
348     Y(L+1)=T
349     JT=K(L)
350     K(L)=K(L+1)
351     K(L+1)=JT
352     ENDP
353     100 CONTINUE
354     N=0
355     DO 101 J=1,6
356     IF(Y(J).NE.0)N=N+1
357     101 CONTINUE
358     RETURN
359     END
360     C
361     C *****SUBROUTINE T2STAT*****
362     C This routine the magnitude of the vectors (arrays) DIFF and RATLOG.
363     C It also finds the largest entry and the subscript of that entry for
364     C each array.
365     C
366     SUBROUTINE T2STAT(DIFF,RATLOG,LMAX,DIFFMAX,RADMAX,
367     *LRMAX,RADIUS,RLOG)
368     DIMENSION DIFF(6),RATLOG(6)
369     RAD=0.
370     RADMAX=0.
371     RADRAT=0.
372     DIFFMAX=0.
373     DO 330 L=1,6
374     RAD=RAD+DIFF(L)*DIFF(L)

```

# Appendix VI

```

375   RADRAT=RADRAT+RATLOG(L)*RATLOG(L)
376   IF(ABS(RATLOG(L)).GT.ABS(RADMAX))THEN
377     RADMAX=RATLOG(L)
378     LRMAX=L
379   ENDDIF
380   IF(ABS(DIFF(L)).GT.ABS(DIFFMAX))THEN
381     DIFFMAX=DIFF(L)
382     LMAX=L
383   ENDDIF
384   330 CONTINUE
385   RADRUS=SQRT(RAD)
386   RLOG=SQRT(RADRAT)
387   C WRITE(8,8800)RADMAX
388   8800 FORMAT('RADMAX',E12.4)
389   RETURN
390   END
391   C
392   C*****SUBROUTINE SORTNK*****
393   C
394   C This routine opens the file containing the values of the index of refraction
395   C as a function of wavelength. Then the imaginary part of the index and the
396   C wavelength that corresponds to it are read in and stored in arrays. The real
397   C part of the index is not saved. The rest of the routine sorts through the
398   C array containing the imaginary part of the index and finds all maxima and
399   C minima. If there is an interval over which there are local extrema and
400   C several points share the extreme value, then the program will write the
401   C extreme value at the two endpoints of the interval, and it is implicitly
402   C understood that k is constant over that interval. Also the first and last
403   C values in the file containing the index of refraction as a function of
404   C wavelength is counted as a maximum or minimum depending on whether
405   C k is increasing or decreasing at those points.
406   C
407   SUBROUTINE SORTNK(PNAME,NMIN,NMAX,WLNMX,WLNMIN,ANKMIN,ANKMAX)
408   CHARACTER*15 PNAME
409   REAL WLNMIN(40),WLNMX(40),ANKMIN(40),ANKMAX(40),WLN(500),NK(500)
410   INTEGER NMIN,NMAX
411   OPEN(UNIT=12,FILE=PNAME)
412   READ(12,*)NPTS
413   DO 300 M=1,NPTS
414     READ(12,*)WLN(M),ANK,NK(M)
415   C WRITE(8,8888)WLN(M),ANK,NK(M)
416   8888 FORMAT(E12.4)
417   300 CONTINUE
418   NMIN=0
419   NMAX=0
420   IF(NK(1).GT.NK(2))THEN
421     NMAX=NMAX+1
422     WLNMX(NMAX)=WLN(1)
423     ANKMAX(NMAX)=NK(1)
424   ELSE
425     NMIN=NMIN+1
426     WLNMIN(NMIN)=WLN(1)
427     ANKMIN(NMIN)=NK(1)
428   ENDDIF

```

# Appendix VI

```

428      M-2
429      99 DO 301 J=M,NPTS-1
430      IF(NK(J).GT.NK(J-1).AND.NK(J).GT.NK(J+1))THEN
431      NMAX=NMAX+1
432      WLNMX(NMAX)=WLN(J)
433      ANKMAX(NMAX)=NK(J)
434      ELSE IF(NK(J).LT.NK(J-1).AND.NK(J).LT.NK(J+1))THEN
435      NMIN=NMIN+1
436      WLNMIN(NMIN)=WLN(J)
437      ANKMIN(NMIN)=NK(J)
438      C
439      C Careful about drawing a false conclusion if two neighboring entries in
440      C the data file are equal.
441      C
442      ELSE IF(NK(J).GT.NK(J-1).AND.NK(J).EQ.NK(J+1))THEN
443      NMAX=NMAX+1
444      WLNMX(NMAX)=WLN(J)
445      ANKMAX(NMAX)=NK(J)
446      JJ=J
447      MM=1
448      DO 302 MMM=J+1,NPTS-1
449      MM=MM+1
450      IF(NK(MMM).GT.NK(MMM+1))THEN
451      NMAX=NMAX+1
452      WLNMX(NMAX)=WLN(MMM)
453      ANKMAX(NMAX)=NK(MMM)
454      GO TO 303
455      ENDP
456      IF(NK(MMM).LT.NK(MMM+1))NMAX=NMAX-1
457      IF(NK(MMM).LT.NK(MMM+1))GO TO 303
458      302 CONTINUE
459      ELSE IF(NK(J).LT.NK(J-1).AND.NK(J).EQ.NK(J+1))THEN
460      NMIN=NMIN+1
461      WLNMIN(NMIN)=WLN(J)
462      ANKMIN(NMIN)=NK(J)
463      JJ=J
464      MM=1
465      DO 305 MMM=J+1,NPTS-1
466      MM=MM+1
467      IF(NK(MMM).LT.NK(MMM+1))THEN
468      NMIN=NMIN+1
469      WLNMIN(NMIN)=WLN(MMM)
470      ANKMIN(NMIN)=NK(MMM)
471      GO TO 303
472      ENDP
473      IF(NK(MMM).GT.NK(MMM+1))NMIN=NMIN-1
474      IF(NK(MMM).GT.NK(MMM+1))GO TO 303
475      305 CONTINUE
476      ENDP
477      301 CONTINUE
478      IF(NK(NPTS).GT.NK(NPTS-1))THEN
479      NMAX=NMAX+1
480      WLNMX(NMAX)=WLN(NPTS)
481      ANKMAX(NMAX)=NK(NPTS)

```

## Appendix VI

```
482     ELSE
483     NMIN=NMIN+1
484     WLNMIN(NMIN)-WLN(NPTS)
485     ANOMIN(NMIN)-NK(NPTS)
486     ENDF
487     GO TO 304
488     303 M-J+MM
489     GO TO 99
490     304 CLOSE(12)
491     RETURN
492     END
```

## Appendix VI

### AVI.1.4 SAMPLE CALCULATIONS: PARAMS AND FILENAMES INPUTS, DATA OUTPUT FILES.

AVI.1.4.1 *This is the "params" file for the composite (soil) surface.*

```
1 TestC1c.c
2 TestC1c.d
3 composite
4 -1 0. 0.
5 .3
6 -1 0. 0.
7 .5
8 71 9.0 .05
9 22 0. 4.
10 0
11 compos.nk
12 .000001 .00001
13 .0001 .005i
```

AVI.1.4.2 *This is the "params" file for the SF96-contaminated surface.*

```
1 TestS1c.c
2 TestS1c.d
3 sf96
4 -1 0. 0.
5 .3
6 -1 0. 0.
7 .5
8 71 9.0 .05
9 22 0. 4.
10 0
11 sf96.nk
12 .000001 .00001
13 .0001 .005
```

AVI.1.4.3 *This is an example of a correctly edited version of FILENAMES.*

```
TestC1c.d
TestS1a.d
testdetect.out
commdetect.out
compos.nk
sf96.nk
1
1
0
```

## AVI.2 DECIDE2

### AVI.2.1 Running DECIDE2.

This section contains a general outline of program DECIDE2. Input and output files are presented on a line-by-line basis. Instructions for running the program and its access to sample input and output files are given at the end of this section.

#### AVI.2.1.1 Functional outline of DECIDE2.

Program DECIDE2 functions in the following manner.

1. Read the input variables.
  - a) Quit and write a message to unit 8 if there is a conflict.
2. Write a heading block to the output file (unit 9).
3. Open and read files containing the indices of refraction for the target and background materials.
  - a) Compute the relative permittivity  $\epsilon_r = (n - ik)^2$  for each material as a function of wavelength.
  - b) Locate and save all resonant wavelengths of the target material.
  - c) Locate and save all resonant wavelengths of the background material.
  - d) Locate and save all wavelengths corresponding to local maxima on the function  $k^t - k^b$ , where  $k^t$  and  $k^b$  are the imaginary parts of the indices of refraction for the target and background materials, respectively.
4. Locate initial wavelengths and examine.
  - a) The first two are the primary resonant wavelengths for each material.
  - b) The third wavelength corresponds to  $(k^t - k^b)_{\max}$ . If this wavelength is equal to either of the first two, the program uses that wavelength that corresponds to the second largest value for a local maximum on  $k^t - k^b$ .
5. For both materials, compute the Mueller matrices as functions of incident angle for given wavelength selections in (4). The following procedure is executed each time a pair of Mueller matrices is computed.
  - a) Interpolate to find  $\epsilon_r$  for each material (do this only once per wavelength).
  - b) Calculate Equations (37) and (39).
  - c) Arrange the terms in Equations (37) and (39) in decreasing order of magnitude.
  - d) Write to unit 9:
    - 1) the subscripts of all non-zero terms in Equation (39);
    - 2) the subscripts of all non-zero terms in Equation (37);
    - 3) Equation (39), largest term in Equation (39), and the subscript of that term;
    - 4) and Equation (37), largest term in Equation (37), and the subscript of that term.
  - e) A check that the user specified limit on the number of Mueller matrices the program is allowed to compute has not been exceeded.
  - f) A check whether the values of Equations (37) or (39) have increased over their previous maximum value.

6. Examine the Mueller elements at adjacent wavelengths and angles computed in (5) for the largest value in Equation (37) or Equation (39). The user must decide which Equation the program bases its decisions on. Adjacent wavelengths and angles are all possible combinations of  $\lambda_0 + j\lambda_{step}$  and  $\theta_0 + k\theta_{step}$ ,  $j, k = -1, 0, 1$ .
  - a) Check to make sure that calculations are not being repeated. That is, do not compute data for the same angle/wavelength pair twice. If data has already been computed for the wavelength/angle pair in question, go on to the next pair.
  - b) Compute  $\epsilon_r$  for each material (do this once per wavelength).
  - c) Compute the Mueller matrices.
    - 1) If the maximum number of Mueller matrices has been exceeded, quit program.
  - d) Calculate Equations (37) and (39).
  - e) Sort terms in Equations (37) and (39).
  - f) Write to unit (9):
    - 1) the subscripts of all non-zero terms in Equation (39);
    - 2) the subscripts of all non-zero terms in Equation (37);
    - 3) Equation (39), largest term in Equation (39), and the subscript of that term;
    - 4) and Equation (37), largest term in Equation (37), and the subscript of that term.
  - g) Check if the values of Equations (37) and (39) are improved.
7. If after completing step 6 an improvement in the results of Equation's (37) or (39) are detected, go back to step 6. If not, proceed to step 8.
8. Write:
  - a) to unit 8 'THE PROGRAM STOPPED BECAUSE NO IMPROVEMENT IN THE RESULTS OF ROUTINE T2STAT HAS BEEN DETECTED';
  - b) and to unit 9 'THE MOST PROMISING ANGLE AND WAVELENGTH PAIR IS' (the wavelength and angle are written next to the statement).
9. The Mueller matrices, at this point, for each material have been stored in arrays. To organize this output, its data is sorted then written in a nested loop where  $\lambda_1 < \lambda_2 < \dots < \lambda_n$  and  $\theta_1 < \theta_2 < \dots < \theta_m$ . The sorted Mueller elements are written to unit 9 in a nested loop where  $\theta$  varies more rapidly than  $\lambda$ .
10. Once an optimum angle/wavelength pair  $\lambda_0, \theta_0$  is found, the program computes and analyzes 121 additional Mueller matrices for the target and background materials. These computations are made for every combination of 11 wavelengths and 11 incident angles. The wavelengths are given by  $\lambda_0 + j \times 0.05$ ,  $j = -5, -4, \dots, 5$ . The incident angles are given by  $\theta_0 + j \times 4$ ,  $j = -5, -4, \dots, 5$ . That is, the wavelength is incremented 11 times in .05  $\mu m$  steps centered about  $\lambda_0$ , and the incident angle is incremented 11 times in 4 degree steps centered about  $\theta_0$ . The results of this part of the program are written to units 10, 11, and 12 in a format that is readable by the plotting package DISPLAY.
  - a) Write the Mueller matrices for the target material to unit 10.
  - b) Write the Mueller matrices for the background material to unit 11.
  - c) Write the results of the Mueller matrix elements analyses to unit 12.
11. Quit the program.

## Appendix VI

Program DECIDE2 needs three input files. The first file, 'DATAIN2,' is used in every run and contains all information needed to control the program. The remaining input files contain indices of refraction of target and background materials as functions of wavelength.

### AVI.2.2.1 DATAIN2.

An example of a correctly edited file is given at the end of this section. The line-by-line format is as follows (for multiple entries per line, separate each entry by spaces):

- A- filename containing the background material's index of refraction as a function of wavelength (up to 15 characters);
- B- filename containing the target material's index of refraction as a function of wavelength (up to 15 characters);
- C- filename for data output, unit 9 (15 characters max);
- D- filename for commentary output, unit 8 (15 characters max);
- E- a descriptive name of the background material (13 characters and no more);
- F- and a descriptive name of the target material (13 characters and no more).
- G- Values of  $\langle h^2 \rangle$  and  $\sigma_s^2$ . These are the mean square height in ( $\mu m^2$ ) and mean square slope, respectively, for the background material. (real, real)
- H- Values of  $\langle h^2 \rangle$  and  $\sigma_s^2$ . These are the mean square height in ( $\mu m^2$ ) and the mean square slope, respectively, for the target material. (real, real)
- I- The number of  $\theta$ 's, with  $\theta_{min}$ , then  $\theta_{step}$ . These values must be in degrees. The program uses this information to fill an array with all incident angles used in the computations. These angles are given by  $\theta_k = \theta_{min} + k \theta_{step}$ ,  $k=0,1,\dots$ , #of  $\theta$ 's -1. (integer, real, real)
- J- The parameters  $\lambda_{min}$ ,  $\lambda_{max}$ ,  $\lambda_{step}$ . These values must be in  $\mu m$ . The program does not consider wavelengths which are  $< \lambda_{min}$  or  $> \lambda_{max}$  in any of its operations. (real, real, real)
- K- The code number RCODEB. It determines which auto-correlation function the program will use for the background material, or to normalize the Mueller matrices to  $f_{11}$ . RCODEB=0 (normalize to  $f_{11}$ ), RCODEB= 1 (Gaussian), RCODEB=2 (N=8), RCODEB=3 (N=6). (integer)
- L- The code number, RCODET. It determines which auto-correlation function the program will use for the target material, or to normalize the Mueller matrices to  $f_{11}$ . RCODET=0 (normalize to  $f_{11}$ ), RCODET= 1 (Gaussian), RCODET=2 (N=8), RCODET=3 (N=6). (integer)
- M- Values AERR1 and RERR1. These values are the error requests for the IMSL routine QDAG. AERR1 is the absolute error, and RERR1 is the relative error. The IMSL routine QDAG computes the full wave quantity Q. (real, real)
- N- Values AERR2 and RERR2. These values are the error requests for the IMSL routine TWODQ. AERR2 is the absolute error, and RERR2 is the relative error. The IMSL routine TWODQ computes the full wave quantity  $\sigma_{ij}^H$  [1]. (real, real)
- O- Value NQUIT. The program will compute no more than NQUIT Mueller matrices. (integer)
- P- Value NCRIT. If NCRIT=1, the program bases its decisions on Equation (37). If NCRIT=0, the program bases its decisions on Equation (39). (integer)

- Q- A file given on this line is open to unit 10. 121 Mueller matrices for the target material are written to this file in a format that is readable by the plotting package DISPLAY. (character 15)
- R- A file given on this line is open to unit 11. 121 Mueller matrices for the background material are written to this file in a format that is readable by the plotting package DISPLAY. (character 15)
- S- A file given on this line is open to unit 12. The results this analyses of 121 Mueller matrices, units 10 and 11, are written to this file in a format that is readable by DISPLAY. (character 15).

#### AVI.2.2.2 Index of refraction files.

For each run, DECIDE2 reads two files of refractive index data. The names of these files are passed to lines A and B of the file DATAIN2. They are required for two reasons: (1) relative permittivity  $\epsilon_r = (n - ik)^2$  is computed by interpolation at every wavelength the program computes Mueller matrices (both  $\lambda$  and  $\epsilon_r$  are passed to the subroutine RETRO); and (2) the index of refraction specifies where resonant wavelengths occur.

The index of refraction files have the following format:

- A- number of wavelengths in the list (integer);
- B-  $\lambda_1, n_1, k_1$  (real, real, real);
- .....
- .....
- Z- and  $\lambda_m, n_m, k_m$  (real, real, real),

where  $\lambda_j < \lambda_{j+1}$ . Also, for the interpolation routine to work,  $m > 6$ . A sample file is shown at the end of this guide.

#### AVI.2.3 Output Files.

Program DECIDE2 creates five output files. The first containing Mueller elements and the second contains commentary remarks. Data written to the third, fourth, and fifth files is in a format readable by the plotting program DISPLAY. The names for these files are the inputs to DECIDE2, found on lines C and D of 'DATAIN2.'

##### AVI.2.3.1 Commentary output file.

This file resides on unit 8. Its name is passed on line D of 'DATAIN2.' Unit 8 receives various values for debugging the program in the event of its failure. Several write statements to unit 8 exist in this program, most have been commented out. If necessary, they may be reinstated by removing the C in the leftmost column of the line the write statement appears. During normal operation DECIDE2 writes descriptive messages to unit 8 for the following reasons.

1. The input variable NCRIT appearing on line P of 'DATAIN2' is  $< 0$  or  $> 2$ , the program writes 'ILLEGAL NCRIT' and then stops.

2. The program will write 'THE PROGRAM STOPPED BECAUSE NO IMPROVEMENT IN THE RESULTS OF THE ROUTINE T2STAT HAS BEEN DETECTED' once it had determined the optimum angle/wavelength pair. When this message is written to unit 8 the run has been successful and the program should quit.
3. In the event that one of the input variables RCODEB or RCODET (lines K and L of 'DATAIN2') is 0 and the other is non-zero the program writes 'ILLEGAL COMBINATION FOR RCODEB AND RCODET' then quits.
4. If either RCODEB or RCODET are negative or greater than 3 the subroutine RETRO will write 'ILLEGAL RCODE' and kill the program.
5. Should the number of Mueller matrices calculated exceed NQUIT (line O of DATAIN2) the program will write 'IF YOU WANT TO COMPUTE MORE MUELLER MATRICES, YOU WILL HAVE TO INCREASE THE INPUT VARIABLE NQUIT' and quits.

#### AVI.2.3.2 Data output file.

This file (unit 9) contains 3 distinct sections; the header block, the analysis of the Mueller matrices, and the Mueller matrix elements as computed by subroutine RETRO. The format of the header block is as follows:

- A- a description of the background and target materials;
- B-  $\langle h^2 \rangle$ ,  $\sigma_b^2$  of the background material;
- C-  $\langle h^2 \rangle$ ,  $\sigma_t^2$  of the target material;
- D-  $\lambda_{\min}$ ,  $\lambda_{\max}$ , and  $\lambda_{\text{step}}$ . The program does not initially compute all wavelengths between  $\lambda_{\min}$  and  $\lambda_{\max}$  in increments of  $\lambda_{\text{step}}$ . It first selects three wavelengths between  $\lambda_{\min}$  and  $\lambda_{\max}$ . These wavelengths are those for which  $k^t$ ,  $k^b$ , and  $k^t - k^b$  are maximum over the interval  $[\lambda_{\min}, \lambda_{\max}]$  (the  $k$ 's are the imaginary parts of the index of refraction for the t-target and b-background materials).
- E- A list of all incident angles used by the program. This program computes and analyzes Mueller matrices for both materials at each of these incident angles for the first three wavelengths it selects. The program finds the wavelength  $\lambda_0$  and angle  $\theta_0$  that resulted in the largest value for the discrimination criterion (see line P of 'DATAIN2'). From there, the program computes data at angles and wavelengths near  $\lambda_0$  and  $\theta_0$  until it finds the optimum angle/wavelength pair (Section AVI.1).
- F- Values RCODEB and RCODET. These values are input on lines K and L of 'DATAIN2.'
- G- Value NCRIT. This value is input on line P of 'DATAIN2.'

The second section in this file contains two similar subsections. Both subsections contain data resulting from the analysis of Mueller matrices. The difference between each subsection is the way the program chooses the wavelengths and incident angles used in subsequent analyses.

In the first subsection of this file data is written in a nested loop where  $\theta$  varies more rapidly than  $\lambda$ . The inner loop is repeated once for each incident angle written on line E of the header block. The outer loop is repeated three times. First, the wavelength of primary resonance for the background material is determined. Second, the wavelength of primary resonance for the target material is obtained. Finally, the third and final time through the outer loop selects the optimum wavelengths as discussed in the previous sections. (See also line D of the header block).

In the second subsection of this file, the program looks for angles and wavelengths near the angle and wavelength in the previous subsection that produced the largest value for Equation (37) (NCRT=1) or Equation (39) (NCRT=0 see line P of 'DATAIN2'). If the program finds an angle/wavelength pair that results in a larger value for Equation (37) or (39) it further searches angles and wavelengths in the neighborhood of that pair. This search goes on until the program notices no improvement in the value of Equation (37) or (39).

The format for both subsections is given below.

- A2- Both incident angle and the wavelength are written. This line reads:  
"WAVELENGTH="λ "INCIDENT ANGLE="θ.
- B2- The subscripts of all non-zero components of  $\bar{y}$  Equation (39) are listed in decreasing order of the magnitude of the component they represent. As an example, suppose  $y_5 > y_1 > y_4$  and  $y_2 = y_3 = y_6 = 0$ . For the  $\bar{y}$  described above, this line would read 5 1 4.
- C2- This line is the same as B2, except it is associated with  $\bar{x}$  Equation (37).
- D2-  $|\bar{x}|$ ,  $|\bar{y}|$ ,  $j$ ,  $(x_j)_{\max}$ ,  $i$ ,  $(y_i)_{\max}$   
where;  
 $|\bar{x}|$  is calculated from Equation (37),  
 $|\bar{y}|$  is calculated from Equation (39),  
 $(x_j)_{\max}$  is the largest component of  $\bar{x}$ ,  
and  $(y_i)_{\max}$  is the largest component of  $\bar{y}$ .
- E2- This program tells the user which incident angle and wavelength resulted in the best discrimination between the background and target materials. The line reads 'THE MOST PROMISING ANGLE AND WAVELENGTH PAIR IS' λ, θ.

DECIDE2 writes all of the Mueller elements computed to the third data block. They are written in a nested loop where θ varies more rapidly than λ. Both θ and λ increase monotonically from their minimum to their maximum value. The format is as follows:

- A3- values λ and θ;
- B3- values  $v_1^i, v_2^i, v_3^i, v_4^i, v_5^i$ , and  $\hat{z}$ ;
- C3- values  $v_1^b, v_2^b, v_3^b, v_4^b, v_5^b$ , and  $v_6^b$ .

#### AVI.2.3.3 Mueller matrix output files.

The third and fourth output files are opened as units 10 and 11, respectively. Units 10 and 11 are also used by the program as the index of refraction files for both materials. The program closes units 10 and 11 after indices of refraction data files are read in, and reopens them according to the names on lines Q and R of DATAIN2. The purpose of these files is to make selections of Mueller elements read by the plot program DISPLAY. The third output file contains Mueller elements for the target material, while the fourth output file contains elements for the background material. Both files have 121 Mueller matrices that are formatted in a way identical to RETRO's output. The wavelengths and incident angles used to compute these data are chosen as described in Section AVI.2.1.1.

When using DISPLAY to plot data in either of these files be aware that only one auto-correlation function was computed, selected by input variables on lines K and L of DATAIN2. Also, be aware here that inputs on lines K and L of DATAIN2 determine if DECIDE2 computes and analyzes  $\frac{f_{ij}}{f_{11}}$  or  $f_{ij}$ .

*AVI.2.3.4 Analysis output file.*

This output file contains results of analyses of data in the third and fourth output files. These data are in a format readable by DISPLAY. When DISPLAY asks which matrix element analysis to plot enter 1 1 for a plot of Equation (37) relating to element  $f_{11}$  and enter 2 1 for a plot of Equation (39), element  $f_{21}$ , and so on.

*AVI.2.4 Instructions for running DECIDE2.*

The steps required to run program DECIDE2 are presented in this section. After the input file 'DATAIN2' has been properly edited and all other input files are loaded, DECIDE2 is ready for execution. DECIDE2 may be run in either batch mode or interactively. The number of  $\theta$  (line I of 'DATAIN2') and  $\lambda_{step}$  (line J of 'DATAIN2') parameters determine run time. As a rule of thumb run the program in the batch mode if the number of  $\theta$ 's  $> 30$  or if  $\lambda_{step} < .02$  over the normal 9-12  $\mu m$  band of wavelengths.

*AVI.2.4.1 Interactive runs.*

Typing "decide2.run" after the prompt will compile, link, and execute the program using data stored in 'DATAIN2.' Executable code becomes stored in "decide2.xqt"; simply type this command to iterate a program run during the same login after changing 'DATAIN2.' Before logging off, remove 'decide2.xqt' for it is a rather large file. The object file 'decide2.o' was removed by 'decide2.run' immediately following linking.

*AVI.2.4.2 Batch runs.*

Typing "examp2dec" after the prompt runs this program in the batch mode using the data stored in 'DATAIN2.' The file "examp2dec" submits the file "decide2.bat" to the batch queue. For this reason, the user must make sure that the file "decide2.bat" is in storage on before typing "examp2dec."

## Appendix VI

### AVI.2.5 DECIDE2 SOURCE CODE.

```

1  C This work was done for the CRDEC on the Aberdeen Proving Grounds
2  C Edgewood Area.
3  C This work was done by S. Mark Haugland under contract DAAD05-89-P-0427.
4  C
5  C The purpose of this program is to locate optimum combinations of
6  C wavelength and incident angle for use in distinguishing between
7  C a contaminated and a dry surface. This program assumes the
8  C contaminant forms an optically thick layer on the background surface.
9  C
10 C DECIDE2 computes and analyzes Mueller matrices every time it calls
11 C the subroutine RETRO. These data are used to distinguish between
12 C the background (base) material and the target (contaminant) material.
13 C Each material's Mueller matrix is a function of incident angle
14 C , wavelength, mean square height, and mean square slope.
15 C DECIDE2 locates the primary resonant wavelength for each material
16 C and the wavelength at which the imaginary part of the index of refraction
17 C for each material differ by the greatest amount. At each of these
18 C wavelengths, DECIDE2 computes Mueller matrices for each material as
19 C a function of incident angle. Immediately following computation,
20 C each pair of Mueller matrices is analyzed (see the user manual for
21 C details). The program identifies the combination of these wavelengths
22 C and incident angles that resulted in the best discrimination between the
23 C two surfaces. Next, angle/wavelength pairs near the one that resulted in the
24 C best discrimination are examined. If there is an increase in the
25 C discrimination at some of these angles, the program remembers the one
26 C that resulted in the greatest increase. Angles near that one are examined
27 C next. This process is repeated until there is no further increase in
28 C discrimination.
29 C
30 C This program uses some IMSL routines. For more information about
31 C the IMSL routines, see the subroutine RETRO.
32 C
33 PROGRAM DECIDE2
34 REAL ANG(100),ANB(500),ANT(500),AKB(500),AKT(500),KDIFF(500)
35 REAL WLN(500),WLNT(500),HMSQB,SIGSB,HMSQT,SIGST,WLN(100,100),
36 *ANG1,ANGINC,WLMAX,WLMIN,WLSTEP,WLNMAXT(40),WLNMIN(40)
37 REAL PFT(6),PFB(6),DIFF(6),RATLOG(6),WLNWLN(100),PT(6,100,100)
38 REAL AKMAXB(40),WLNMAXD(40),WLNMIN(40),AKMIN(40),AKMAXD(40)
39 REAL AKMIN(40),AKMAXT(40),WLNMAXB(40),WLNMINB(40),AKMINB(40)
40 REAL FB(6,100,100),ARR(6)
41 INTEGER RCODEB,RCODET,NPTSB,NPTST,NANGMX,KKA(6)
42 COMPLEX ERB(500),ERT(500),ERESTT,ERESTB,CNT(500),CNB(500)
43 COMPLEX NKTEST,NKBEST
44 CHARACTER*13 OUTNM, COMNM, BACKNK, TARGNK, TARGMTX, BACKMTX, ANALMTX
45 CHARACTER*13 BACKMAT,TARGMAT
46 COMMON/TWO/WLMIN,WLMAX
47 C
48 C Read names of input and output files and open them
49 C
50 OPEN(UNIT=7,FILE='DATAIN2')
51 READ(7, '(A)')BACKNK

```

## Appendix VI

```

52      OPEN(UNIT=10,FILE=BACKNK)
53      READ(7,'(A)')TARGNK
54      OPEN(UNIT=11,FILE=TARGNK)
55      READ(7,'(A)')OUTNM
56      OPEN(UNIT=9,FILE=OUTNM)
57      READ(7,'(A)')COMNM
58      OPEN(UNIT=8,FILE=COMNM)
59      C
60      PI=ABS(ATAN2(0,-1.))
61      C
62      C Read names of materials to be examined, and information needed to
63      C characterize them.
64      C
65      READ(7,'(A)')BACKMAT
66      READ(7,'(A)')TARGMAT
67      READ(7,'(A)')HMSQB,SIGSB
68      READ(7,'(A)')HMSQT,SIGST
69      READ(7,'(A)')NANGMX,ANG1,ANGINC
70      IF(NANGMX.LT.0)THEN
71          NANGMX=-NANGMX
72      READ(7,'(A)')ANG(IL),IL-1,NANGMX
73      ELSE
74          DO 40 IL=1,NANGMX
75              40 ANG(IL)=ANG1+(IL-1)*ANGINC
76      ENDDO
77      READ(7,'(A)')WLMIN,WLMAX,WLSTEP
78      WRITE(9,9002)BACKMAT,TARGMAT
79      9002 FORMAT(2A15)
80      WRITE(9,'(A)')HMSQB,SIGSB
81      WRITE(9,'(A)')HMSQT,SIGST
82      WRITE(9,9000)WLMIN,WLMAX,WLSTEP
83      9000 FORMAT(5E12.4)
84      WRITE(9,9000)(ANG(IL),IL-1,NANGMX)
85      C
86      C RCODEB and RCODET are the code variables that set the auto-correlation function
87      C for the background and target materials respectively. It is allowed for the
88      C target and background materials to have different auto-corr functions. RCODEB,
89      C T-1—>Gaussian, -2—>N-8, -3—>N-6. There is an option to normalize
90      C all mtr. elements to P11 this will be the case if both RCODEB and RCODET=0
91      C Thus both RCODEB and RCODET equaling 0 causes the program to analyze Pij/P11
92      C rather than Pij.
93      C
94      READ(7,'(A)')RCODEB
95      READ(7,'(A)')RCODET
96      IF(RCODEB.EQ.0.AND.RCODET.NE.0)GO TO 991
97      IF(RCODET.EQ.0.AND.RCODEB.NE.0)GO TO 991
98      WRITE(9,9003)RCODEB,RCODET
99      9003 FORMAT('CODE FOR AUTO-CORR FUNCTIONS AND NORMALIZATION', 2I5)
100     C
101     C Read the error controls for the integration routines. AERR1 and RERR1
102     C are for QDAG and AERR2 and RERR2 are for TWODQ.
103     C
104     READ(7,'(A)')AERR1,RERR1
105     READ(7,'(A)')AERR2,RERR2

```

## Appendix VI

```

105  C
106  C NCRIT-1 -> decisions are made based on the
107  C Euclidean distance between the Mueller matrix for the target and the Muller
108  C matrix for the background material. NCRIT-0-> decisions are made based on
109  C a test which is independent of the size variations in Muller matrix elements
110  C in the same Mueller matrix. NQUIT is the maximum number of Muller matrices
111  C you will this program to compute in a single run.
112  C
113      READ(7,*)NQUIT
114      READ(7,*)NCRIT
115      WRITE(9,9001)NCRIT
116      9001 FORMAT('NCRIT',I5)
117      IF(NCRIT.LT.0.OR.NCRIT.GT.2)THEN
118          WRITE(8,8767)
119          8767 FORMAT('ILLEGAL NCRIT')
120      STOP
121  ENDP
122  C
123  C The files TARGMTX and BACKMTX are written Mueller matrices for the
124  C target and background materials. These matrices are written in a
125  C way DISPLAY can plot them. Once the optimum angle and wavelength
126  C has been identified, the program computes a rectangle of Mueller matrices
127  C centered at that point. The rectangle contains a range of 40 degrees in 4
128  C degree intervals for theta, and .5 microns in .05 micron intervals for
129  C lambda. These matrices are analyzed by the routine TZSTAT and these
130  C results are written ANALMTX in a form that they can be plotted by
131  C DISPLAY.
132  C
133  C Open TARGMTX, and BACKMTX as units 10 and 11 respectively. Do this after
134  C 10 and 11 have been closed by the part of the program that reads
135  C the indices of refraction in. Open ANALMTX as unit 12.
136      READ(7,*(A))TARGMTX
137      READ(7,*(A))BACKMTX
138      READ(7,*(A))ANALMTX
139  C
140  C Read in the index of refraction for the background material and compute the
141  C relative permittivity.
142  C
143      READ(10,*)NPTS8
144      DO 41 IL=1,NPTS8
145          READ(10,*)WLN8(IL),ANB(IL),AKB(IL)
146          X=ANB(IL)
147          Y=AKB(IL)
148          CNB(IL)=(1.-0.*X+(0.-1.*Y
149          ERB(IL)=(1.-0.*X*X-Y*Y)*(0.-1.*ABS(2.*X*Y)
150  C  WRITE(8,8001)WLN8(IL),CNB(IL)
151      41 CONTINUE
152      CLOSE(10)
153  C
154  C Read in the index of refraction for the target material and compute the
155  C relative permittivity.
156  C
157      READ(11,*)NPTST
158      DO 42 IL=1,NPTST

```

# Appendix VI

```

159     READ(11,*)WLNT(IL),ANT(IL),AKT(IL)
160     X=ANT(IL)
161     Y=AKT(IL)
162     ERT(IL)=(1.0)*X*X-Y*Y*(0.1)*ABS(2.*X*Y)
163     2121 FORMAT(3E12.4)
164     CNT(IL)=(1.0)*X+(0.1)*Y
165     C WRITE(8,8001)WLNT(IL),CNT(IL)
166     42 CONTINUE
167     CLOSE(11)
168     C
169     C Find all local maxima and minima in the absolute value of the imaginary
      C part of the index of refraction for each material.
170     C
171     CALL SORTNK(NMINT, NMAXT, WLNMAXT, WLNMIN, AKMINT, AKMAXT, NPTST,
172     *AKT,WLNT)
173     CALL SORTNK(NMINB, NMAXB, WLNMAXB, WLNMINB, AKMINB, AKMAXB, NPTSB,
174     *AKB,WLNB)
175     C
176     C Find all local maxima and minima in the difference of k for the target
177     C material and k for the background material. There is a technical
178     C problem if the optical constants of each material are not known at the
179     C same wavelengths. This is handled by interpolating the function that
180     C is known at fewer wavelengths at all wavelengths for which the other set
181     C of optical constants is known.
182     C
183     C WRITE(8,*)NPTSB
184     IF(NPTSB.GT.NPTST)THEN
185     DO 43 I=1,NPTSB
186     WLEN=WLNB(I)
187     IF(WLEN.GT.WLNT(NPTST))GO TO 43
188     CALL ERCMP(WLNT,CNT,WLEN,NKTEST)
189     KDIFF(I)=ABS(AIMAG(NKTEST))-AKB(I)
190     C IF(ABS(NKTEST).GT.ABS(CNB(I)))KDIFF(I)=ABS(NKTEST-CNB(I))
191     C WRITE(8,8812),WLEN,KDIFF(I),NKTEST
192     C WRITE(8,8812),WLEN,NKTEST
193     8812 FORMAT(15,5E12.4)
194     43 CONTINUE
195     CALL SORTNK(NMIND, NMAXD, WLNMAXD, WLNMIN, AKMIND, AKMAXD, NPTSD,
196     *KDIFF,WLNB)
197     ELSE
198     DO 44 I=1,NPTST
199     WLEN=WLNT(I)
200     IF(WLEN.GT.WLNB(NPTSB))GO TO 44
201     CALL ERCMP(WLNB,CNB,WLEN,NKBEST)
202     KDIFF(I)=AKT(I)-ABS(AIMAG(NKBEST))
203     C IF(ABS(NKBEST).LT.ABS(CNT(I)))KDIFF(I)=ABS(NKBEST-CNT(I))
204     C WRITE(8,8812),WLEN,KDIFF(I),NKBEST
205     44 CONTINUE
206     CALL SORTNK(NMIND, NMAXD, WLNMAXD, WLNMIN, AKMIND, AKMAXD, NPTSD,
207     *KDIFF,WLNT)
208     ENDP
209     C DO 444 NN=1,NMAXD
210     C WRITE(8,8009)NN,WLNMIND(NN),AKMIND(NN)
211     C 444 CONTINUE

```

# Appendix VI

```

212 C DO 445 NN-1,NMINT
213 C WRITE(8,8009)WLNMIN(NN),AKMIN(NN)
214 C 445 CONTINUE
215 C
216 C Start computing Mueller matrices. Start with the wavelengths that correspond
217 C to the primary resonance for each material. Also look at the wavelength
218 C for which the difference between the imaginary part of the index of refraction
219 C for the target material and that for the base material is at a maximum.
220 C
221 WLNWLN(1)=WLNMAXB(1)
222 IF(WLNWLN(1).NE.WLNMAXT(1))THEN
223 WLNWLN(2)=WLNMAXT(1)
224 ELSE
225 WLNWLN(2)=WLNMAXT(2)
226 ENDIF
227 JK=2
228 DO 45 J=1,NMAXD
229 J=J
230 IF(WLNMAXD(J).LT.1.E-05)GO TO 45
231 IF(AKMAXD(J).LT.0.)GO TO 45
232 IF(WLNWLN(2).NE.WLNMAXD(J).AND.WLNWLN(1).
233 *NE.WLNMAXD(J))THEN
234 JK=3
235 WLNWLN(3)=WLNMAXD(J)
236 GO TO 46
237 ENDIF
238 45 CONTINUE
239 46 MWLN=0
240 MATCNT=0
241 T2STMX=0.
242 RADMX=0.
243 DO 89 J=1,JK
244 MWLN=MWLN+1
245 WLN(J,1)=WLNWLN(J)
246 CALL ERCP(WLNB,ERB,WLN(J,1),ERESTB)
247 C PRINT 1111,WLN(J,1),ERESTB
248 1111 FORMAT(2E12.4)
249 CALL ERCP(WLNT,ERT,WLN(J,1),ERESTT)
250 C WRITE(8,8001)WLN(J,1),ERESTT,ERESTB
251 8001 FORMAT('wavelength index ',5E12.4)
252 DO 89 I=1,NANGMX
253 WLN(J,I)=WLN(J,1)
254 C WRITE(8,*)WLN(J,I),ANG(I)
255 WRITE(9,9569)WLN(J,I),ANG(I)
256 9569 FORMAT('WAVELENGTH- ',E14.6,' INCIDENT ANGLE- ',E14.6)
257 CALL RETRO(ERESTT,HMSQT,SIGST,WLN(J,I),ANG(I),AERR1,RERR1,AERR2
258 *,RERR2,RCODET,PFT,P)
259 CALL RETRO(ERESTB,HMSQB,SIGSB,WLN(J,I),ANG(I),AERR1,RERR1,AERR2
260 *,RERR2,RCODEB,PPB,P)
261 9200 FORMAT('PFT OVER PPB',6E10.4)
262 MATCNT=MATCNT+2
263 C
264 C Save and analyze results from retro.
265 C

```

# Appendix VI

```

266      DO 78 K=-1,6
267      FT(K,J,1)=FFT(K)
268      FB(K,J,1)=FPB(K)
269      78 CONTINUE
270      CALL T2STAT(DIFF,RATLOG,LMAX,DIFFMAX,RADMAX,
271      *LRMAX,RADIUS,RLOG,FFT,FPB)
272      CALL AMTXELE(RATLOG,ARR,KKA,NOP)
273      WRITE(9,9099)(KKA(NNB),NNB=1,NOP)
274      CALL AMTXELE(DIFF,ARR,KKA,NOP)
275      WRITE(9,9099)(KKA(NNB),NNB=1,NOP)
276      WRITE(9,9098)RADIUS,RLOG,LMAX,DIFFMAX,LRMAX,RADMAX
277      9098 FORMAT(2E12.4,I10.4,E12.4,I10.4,E12.4)
278      9099 FORMAT(6D3)
279      8019 FORMAT(2E12.4)
280      8003 FORMAT('PP',E12.4)
281      CALL SAVE(NCRIT,T2STMX,RADMX,J,I,ISMX,IMX,ISRDMX,IRDMX,
282      *JWLSET,RADIUS,RLOG)
283      IF(MATCNT.GE.NQUIT)GO TO 999
284      89 CONTINUE
285      777 IF(NCRIT.EQ.0)DSET=ISRDMX
286      IF(NCRIT.EQ.1)DSET=ISMX
287      IF(NCRIT.EQ.0)MSET=IRDMX
288      IF(NCRIT.EQ.1)MSET=IMX
289      C WRITE(8,8121)DSET,MSET,WLN(DSET,MSET),ANG(MSET)
290      8121 FORMAT('DSET,MSET',2H,2E12.4)
291      C WRITE(8,8122)T2STMX,RADMX
292      8122 FORMAT('RADMX RLOG',2E12.4)
293      C
294      C Look at wavelengths and angles of incidence close to those for which
295      C things look most promising.
296      C
297      KILL=1
298      DO 79 J=-1,1
299      W=WLNWLN(DSET)+J*WLSTEP
300      IF(W.LT.WLMIN)W=WLMIN
301      IF(W.GT.WLMAX)W=WLMAX
302      CALL ERCMP(WLNB,ERB,W,ERESTB)
303      CALL ERCMP(WLNT,ERT,W,ERESTT)
304      DO 80 K=-1,1
305      IF(J.EQ.0.AND.K.EQ.0)GO TO 80
306      KKK=K+MSET
307      IF(KKK.LE.0.OR.KKK.GT.NANGMX)GO TO 80
308      CALL WLANGSET(W,MWLN,WLNWLN,WLN,J,K,DSET,MSET,NCALC,D,M)
309      C WRITE(8,'*NCALC
310      IF(NCALC.EQ.2)GO TO 80
311      C WRITE(8,'*Wln-',WLN(D,M),ANG(M)
312      WRITE(9,9969)WLN(D,M),ANG(M)
313      CALL RETRO(ERESTT,HMSQT,SIGST,WLN(D,M),ANG(M),AERR1,RERR1,
314      *AERR2,RERR2,RCODET,FFT,FI)
315      CALL RETRO(ERESTB,HMSQB,SIGSB,WLN(D,M),ANG(M),AERR1,RERR1,
316      *AERR2,RERR2,RCODEB,FPB,FI)
317      MATCNT=MATCNT+2
318      C
319      C Save and analyze results from retro.

```

# Appendix VI

```

320 C
321 DO 791 KG-1,6
322 FT(KG,D,4)-FFT(KG)
323 FB(KG,D,4)-FPB(KG)
324 791 CONTINUE
325 CALL T2STAT(DIFF,RATLOG,LMAX,DIFFMAX,RADMAX,
326 *LRMAX,RADIUS,RLOG,FFT,FPB)
327 CALL AMTXELE(RATLOG,ARR,KKA,NOP)
328 WRITE(9,9099)(KKA(NNB),NNB-1,NOP)
329 CALL AMTXELE(DIFF,ARR,KKA,NOP)
330 WRITE(9,9099)(KKA(NNB),NNB-1,NOP)
331 WRITE(9,9098)RADIUS,RLOG,LMAX,DIFFMAX,LRMAX,RADMAX
332 CALL SAVE(NCRIT,T2STMX,RADMX,D,4,DMX,4MX,BRDMX,4RDMX,
333 *JWLSET,RADIUS,RLOG)
334 IF(JWLSET.EQ.2)KILL-2
335 C WRITE(9,'*Kill,match ',KILL,MATCH)
336 80 CONTINUE
337 79 CONTINUE
338 IF(KILL.EQ.1)GO TO 996
339 IF(MATCH.GE.NQUIT)GO TO 999
340 GO TO 777
341 996 WRITE(8,'*PROGRAM STOPPED BECAUSE NO IMPROVEMENT IN THE'
342 WRITE(8,'*RESULTS OF ROUTINE T2STAT HAS BEEN DETECTED'
343 GO TO 998
344 991 WRITE(8,8000)
345 8000 FORMAT('ILLEGAL COMBINATION FOR RCODEB AND RCODET')
346 GO TO 1090
347 999 IF(MATCH.GE.NQUIT)WRITE(8,8080)
348 8080 FORMAT('IF YOU WANT TO COMPUTE MORE MATRICES YOU WILL HAVE TO'
349 *',' INCREASE THE INPUT VARIABLE NQUIT')
350 C
351 C Arrange the Mueller matrices so that they can be written to unit 9
352 C in a nested loop in which ANG(4) varies more rapidly than WLNWLN(3).
353 C
354 998 WRITE(9,9013)WLN(3SET,4SET),ANG(4SET)
355 9013 FORMAT('THE MOST PROMISING WAVELENGTH AND ANGLE PAIR IS',2E12.4)
356 DO 98 J-1,MWLN-1
357 DO 98 D-1,MWLN-J
358 IF(WLNWLN(D).GT.WLNWLN(D+1))THEN
359 T1-WLNWLN(D)
360 WLNWLN(D)-WLNWLN(D+1)
361 WLNWLN(D+1)-T1
362 DO 111 M-1,NANGMX
363 T1-WLN(D,M)
364 WLN(D,M)-WLN(D+1,M)
365 WLN(D+1,M)-T1
366 DO 111 K-1,6
367 T1-FT(K,D,M)
368 FT(K,D,M)-FT(K,D+1,M)
369 FT(K,D+1,M)-T1
370 T1-FB(K,D,M)
371 FB(K,D,M)-FB(K,D+1,M)
372 FB(K,D+1,M)-T1
373 111 CONTINUE

```

# Appendix VI

```

374     ENDIF
375 98 CONTINUE
376 C Write Mueller Matrices for each material to unit (9)
377   DO 112 D=1,MWLN
378     DO 112 M=1,NANGMX
379       IF(WLN(D,M).GT.1.E-05)THEN
380         WRITE(9,9999)WLNWLN(D),ANG(M)
381         9999 FORMAT(2E14.6)
382         WRITE(9,9998)(PT(K,D,M),K=1,6)
383         WRITE(9,9998)(PB(K,D,M),K=1,6)
384         9998 FORMAT(6E12.4)
385       ENDIF
386   112 CONTINUE
387 C Determine the 11 wavelengths and 11 angles that the program computes
388 C and analyzes Mueller matrices for. These matrices and the results of
389 C the analysis are written to units 10, 11, and 12.
390   CALL SETHETWL(ANG,WLNWLN,DSET,MSET)
391   OPEN(UNIT=10,FILE=TARGMTX)
392   OPEN(UNIT=11,FILE=BACKMTX)
393   OPEN(UNIT=12,FILE=ANALMTX)
394 C Write heading blocks to units 10, 11, and 12 so DISPLAY can read the
395 C data contained in those files.
396   WRITE(10,'(A)')TARGMAT
397   WRITE(11,'(A)')BACKMAT
398   WRITE(12,'(A)')BACKMAT
399   NH=1
400   NS=1
401   NANG=11
402   NWLEN=11
403   WRITE(10,1000)NH,NS,NWLEN,NANG
404   WRITE(11,1000)NH,NS,NWLEN,NANG
405   WRITE(12,1000)NH,NS,NWLEN,NANG
406   1000 FORMAT(I10.4)
407   WRITE(10,1001)HMSQT
408   WRITE(10,1001)SIGST
409   WRITE(10,1001)(WLNWLN(D),D=1,11)
410   WRITE(10,1001)(ANG(M),M=1,11)
411   WRITE(11,1001)HMSQB
412   WRITE(11,1001)SIGSB
413   WRITE(11,1001)(WLNWLN(D),D=1,11)
414   WRITE(11,1001)(ANG(M),M=1,11)
415   WRITE(12,1001)HMSQB
416   WRITE(12,1001)SIGSB
417   WRITE(12,1001)(WLNWLN(D),D=1,11)
418   WRITE(12,1001)(ANG(M),M=1,11)
419   1001 FORMAT(5E12.4)
420 C Compute and analyze the matrices.
421   DO 113 D=1,11
422     W=WLNWLN(D)
423     CALL ERCMP(WLNB,ERB,W,ERESTB)
424     CALL ERCMP(WLNT,ERT,W,ERESTT)
425   113 M=1,11
426   CALL RETRO(ERESTT,HMSQT,SIGST,W,ANG(M),AERR1,RERR1,
427     *AERR2,RERR2,RCODET,PFT,P)

```

# Appendix VI

```

428 CALL RETRO(ERESTB,HMSQB,SIGSB,W,ANG(4),AERR1,RERR1,
429 *AERR2,RERR2,RCODEB,PPB,PI)
430 CALL T2STAT(DIFF,RATLOG,LMAX,DIFFMAX,RADMAX,
431 *LRMAX,RADIUS,RLOG,PFT,PPB)
432 Q1=1.
433 Q2=1.
434 Q3=1.
435 WRITE(10,1002)Q1,Q2,Q3
436 WRITE(11,1002)Q1,Q2,Q3
437 WRITE(12,1002)Q1,Q2,Q3
438 WRITE(10,1002)(PFT(I),I=1,6)
439 WRITE(11,1002)(PPB(I),I=1,6)
440 WRITE(12,1002)RADIUS,RLOG,DIFFMAX,RADMAX,DIFF(1),RATLOG(1)
441 1002 FORMAT(6E12.4)
442 113 CONTINUE
443 CLOSE(7)
444 CLOSE(8)
445 CLOSE(9)
446 CLOSE(10)
447 CLOSE(11)
448 CLOSE(12)
449 1090 STOP
450 END
451 C
452 C-----SUBROUTINE ERCMP-----
453 C
454 C THIS SUBROUTINE COMPUTES THE VALUE ER FOR USE IN RETRO.
455 C Since this routine interpolates complex functions, it is also used
456 C to interpolate the index of refraction. This routine is a function in the
457 C original version of Retro and was taken of that program and modified for use
458 C here. The argument list is as follows:
459 C     WLN- An array containing the wavelengths at which the index of
460 C         refraction is known. This information was read in from unit
461 C         10 or 11.
462 C     ER- This is a complex array containing the relative permittivities
463 C         as a function of wavelength. All wavelengths are in microns.
464 C     WLEN-The wavelength in microns for which the relative permittivity is
465 C         desired.
466 C     EREST-The relative permittivity returned to the main program.
467 C
468 SUBROUTINE ERCMP(WN,ER,WLEN,EREST)
469 INTEGER NPTS,DEG,I,J,MN,MAX,DDEG,DNPTS
470 REAL WN(300)
471 REAL WLEN,WNP,FACTOR,LI
472 COMPLEX ER(300),EREST
473 C
474 C this converts wavelength in microns to wave number in 1/cm. COMMENT
475 C OUT AND WORK WITH THE WAVELENGTH RATHER THAN THE WAVENUMBER!!!!!!
476 C WNP=10000./WLEN
477 WNP=WLEN
478 DEG=3
479 I=(DEG+1)/2
480 C WRITE(8,8002)WLEN
481 IF(WN(I).GT.WNP) THEN

```

# Appendix VI

```

482     MIN-1
483     MAX-MIN+DEG
484     GOTO 211
485     ENDP
486     200 IF(WN(I).GT.WNP) GOTO 210
487     I-I+1
488     IF(I.EQ.NPTS-(DEG+1)/2)THEN
489     MAX-NPTS
490     MIN-NPTS-DEG
491     GOTO 211
492     ENDP
493     GOTO 200
494     C
495     210 MIN-I-DEG/2-1
496     MAX-MIN+DEG
497     211 FACTOR - 1.
498     C WRITE(I,"YMIN,MAX ",MIN,MAX
499     C
500     8002 FORMAT(E12.4)
501     DO 220 J= MIN,MAX
502     IF(WNP.NE.WN(I))GOTO 220
503     EREST-ER(I)
504     RETURN
505     220 FACTOR-FACTOR*(WNP-WN(I))
506     C
507     EREST-(0.,0.)
508     DO 230 I=MIN,MAX
509     LI-FACTOR/(WNP-WN(I))
510     DO 240 J= MIN,MAX
511     240 IF(I.NE.J)LI-LI*(WN(I)-WN(J))
512     230 EREST-EREST+ER(I)*LI
513     RETURN
514     END
515     C-----
516     C-----SUBROUTINE RETRO-----
517     C
518     C THIS WORK WAS ALTERED BY S.M. HAUGLAND ON 6-23-89.
519     C THE REVISIONS TO RETRO ALLOW IT TO BE CALLED BY ANOTHER PROGRAM
520     C AS A SUBROUTINE.
521     C
522     c This work was done for the CRDEC on the Aberdeen Proving Grounds
523     c Edgewood Area.
524     c This work was done by Craig M. Herzinger under contract 88MQ450.
525     c
526     c This program uses the Full-wave Theory for computing the scattering
527     c of a plane wave from a randomly rough surface.
528     c
529     c This program is for SINGLE scatter from an ISOTROPIC rough surface.
530     c This program is for BACKSCATTER only.
531     c This program is for DIFFUSE scattering only.
532     c
533     c Single scatter implies the reflected radiation struck the rough
534     c surface only one time and that multiple scattering is unaccounted for.
535     c An isotropic surface is considered to invariant to rotation and translation

```

## Appendix VI

536 c in terms of the average scattering.  
 537 c Backscatter implies the receiver and detector are at the same point,  
 538 c with the same orientation, at a point far from the surface.  
 539 c Only diffuse scattering is calculated because the coherent specular term  
 540 c which occurs at normal incidence for backscatter is dropped.  
 541 c  
 542 c This program calculates the 8 generally non-zero Mueller Mtx elements,  
 543 c P11,12,21,22,33,34,43, and 44, for use with the standard Stokes Vector  
 544 c notation. Of these eight two pairs are degenerate, P12-P21 and P34-P43.  
 545 c The elements are calculated on a per solid angle basis so that the calculated  
 546 c Mueller Mtx is absolutely correct to within a scalar constant. The scalar  
 547 c constant is based upon the size of the solid angle intercepted by the  
 548 c detector for a particular experimental setup. For this work to be valid  
 549 c the detector must look at range of returned angles close enough to  
 550 c pure backscatter that the backscattered return is a good approximation  
 551 c of the entire reflected range. Also, the solid angle intercepted by the  
 552 c detector must be invariant to incident angle and wavelength.  
 553 c  
 554 c The program first calculates the scattering mtx, S, for use with the  
 555 c modified Stokes Vector notation, and this is then transformed into the  
 556 c desired form.  
 557 c  
 558 c The elements of S can be written as a product of two values, Q, and  
 559 c a 2-d slope averaging integral. This is allowed by the Full-wave  
 560 c Theory ONLY because slopes and heights are considered uncorrelated.  
 561 c Q is a function of incident angle, surface height auto-correlation  
 562 c function, and free space radiation wavelength. The slope integral  
 563 c is function of incident angle, mean squared slope, and wavelength  
 564 c through the surfaces relative dielectric constant.  
 565 c  
 566 c For the assumed isotropic surface, Q, normally a 2-d integral can  
 567 c be rewritten in polar coordinates and transformed into a 1-d integral.  
 568 c Q<sub>i</sub> is computed by subroutine QCMP for 3 different spectral density/  
 569 c surface height auto-correlation functions. The inputs for the auto-corr.  
 570 c functions are mean squared height and mean squared slope. The 3  
 571 c functions are Gaussian, N-8, and N-6.  
 572 c  
 573 c The slope averaging integrals account for all possible combinations of  
 574 c slopes in the x and z directions. Considering various polarizations and  
 575 c phase relationships, 16 unique integrals are possible to complete S.  
 576 c But 8 integrals → 0 because the integrand is odd.  
 577 c 3 others converge to 1 value, and another → 0 because the integrand  
 578 c is proportional to the imaginary part of a real number.  
 579 c This leaves 5 unique 2-d integrals performed by subroutine IDDCMP.  
 580 c The 5 values could be used compute S but they are recombined in one  
 581 c step to form the Mueller mtx elements, Pij.  
 582 c  
 583 c The following version 10.0 BMSL routines are required:  
 584 c QDAG, TWODQ, BSJO, BSK0, BSK1, and ERPC.  
 585 c These are available from BMSL, customer relations, sixth floor,  
 586 c NBC Building 7300 Bellaire Boulevard, Houston, Texas 77036-5085, USA.  
 587 c Telephone (713)772-7927 Telex: 79-1923 BMSL INC HOU  
 588 c  
 589 c

## Appendix VI

```

590      c
591      c Unit 8 is sent various values for debugging the program and determining
592      c where something may have gone wrong. It sends values for monitoring the
593      c integration routines. Most writes to unit 8 have been commented out, but
594      c can be reinstated should the program have run time problems.
595      c
596      C
597      SUBROUTINE RETRO(EREST, HHMSQ, SSIGS, WWLEN, ANG, AERR1, RERR1, AERR2
598      *, RERR2, RCODE, PP, PPI)
599      REAL PI
600      1  HMSQ, CLEN, WLEN, CLENSQ, SIGS, KO, KOSQ, CSTHT, SNTHT, THETA,
601      3  IDD(16), P(16), PP(6), Q(5),
602      4  AERR1, RERR1, AERR2, RERR2
603      C Don't use SVIDD because as a called routine RETRO does not remember
604      C what the last set of parameters input to it were. The main program
605      C does this.
606      C REAL SVIDD(100,16)
607      C
608      COMPLEX ER,EREST
609      C
610      INTEGER IL,DCODE,RCODE,IFLAG1
611      COMMON/ONE/HMSQ,CLEN,CLENSQ,WLEN,THETA,SNTHT,CSTHT,SIGS,KO,PI
612      C
613      C
614      C THE DESIRED PARAMETERS ARE USED TO COMPUTE THE MUELLER
615      C MATRIX ELEMENTS BY FIRST CALCULATING A SCALING 1-D INTEGRAL, Q,
616      C AND THEN CALCULATING 16 2-D INTEGRALS, IDD1..IDD16, THAT ARE COMBINED IN THE
617      C CORRECT MANNER TO GIVE THE MUELLER MATRIX ELEMENTS FOR THE
618      C STANDARD STOKES VECTOR NOTATION.
619      C IN REALITY FOR THIS WORK ONLY 5 2-D INTEGRALS NEED BE CALCULATED BUT
620      C THE PROGRAM IS SET UP GENERALLY.
621      C The correlation length can be calculated when hmaq and sigs are fixed
622      HMSQ=HHMSQ
623      WLEN=WWLEN
624      PI=PPI
625      SIGS=SSIGS
626      CLENSQ=4.*HMSQ/SIGS
627      CLEN=SQRT(CLENSQ)
628      C PRINT *,HMSQ,SIGS,WLEN,ANG,RCODE,PI
629      C
630      C IOLD=0
631      KO=2*PI/WLEN
632      KOSQ=KO*KO
633      C write(8,*)'Relative dielectric constant ',ER
634      C
635      C Theta is the angle between the incident direction and the normal to the
636      C reference plane in radians
637      THETA=PP*ANG/180.0
638      CSTHT=COS(THETA)
639      SNTHT=SIN(THETA)
640      C
641      C write(8,*)'Angle(deg) Wlen Mean Sq Slope Mean Sq Hgt'
642      C write(8,*)ANG,WLEN,SIGS,HMSQ
643      C

```

# Appendix VI

```

644 C
645 C THIS SECTION COMPUTES THE SCALAR Q VALUES FOR 3 AUTO-CORR FUNCTIONS
646 C RCODE-1 -> GAUSSIAN RCODE-2 -> N-8 RCODE-3 -> N-6 RCODE-0->
647 C YOU ARE NORMALIZING TO P11 AND DO NOT NEED A Q VALUE BECAUSE Q IS THE
648 C SAME FOR ALL MATRIX ELEMENTS.
649 C IF ALL Q VALUES ARE TREATED AS 0 THEN IFLAG1-0 AND THE IDD'S DO NOT
650 C NEED TO BE CALCULATED.
651 IFLAG1-0
652 IF(RCODE.NE.0)CALL QCMP(Q(RCODE),AERR1,RERR1,RCODE)
653 IF(RCODE.EQ.0)Q(RCODE)=1.
654 IF(Q(RCODE).GT.0.)IFLAG1-1
655 C
656 C
657 C THIS SECTION COMPUTES THE 16 IDD VALUES NEEDED FOR THE MUELLER MTX.
658 C NOTE ONLY 5 DISTINCT INTEGRATIONS ARE DONE. THE OTHERS ARE KNOWN FOR
659 C OTHER REASONS ASSIGNED TO THE FOLLOWING CONDITIONAL ASSIGNMENTS.
660 C IF IFLAG1-1 THEN THERE IS A REASON TO CALCULATE THESE VALUES
661 C IF Er IS HELD CONSTANT OVER A RANGE OF WAVELENGTHS THEN IDD'S ONLY
662 C NEED TO BE CALCULATED ONCE FOR EACH INCIDENT ANGLE.
663 C
664 C Set DCODE equal to zero so as not to use SVIDD
665 DCODE=0
666 IOLD=0
667 C
668 IF(IFLAG1.GT.0.) THEN
669 C IF(DCODE.NE.1.OR.IOLD.NE.1) THEN
670 DO 90 IL-1,16
671 IF(IL.EQ.1.OR.IL.EQ.2.OR.IL.EQ.4.OR.IL.EQ.5.OR.IL.EQ.6)THEN
672 CALL IDDCMP(IDD(IL),AERR2,RERR2,IL,EREST)
673 ELSE IF(IL.EQ.3.OR.IL.EQ.7) THEN
674 IDD(IL)=IDD(2)
675 ELSE IF(IL.EQ.8) THEN
676 IDD(IL)=0.
677 ELSE IF(IL.GE.8) THEN
678 IDD(IL)=0.
679 ENDIF
680 90 CONTINUE
681 C If Er is constant then save IDD's for next wavelength
682 C IF(DCODE.EQ.1) THEN
683 DO 93 IL-1,16
684 C SVIDD(14,IL)=IDD(IL)
685 C 93 CONTINUE
686 C ENDP
687 C ELSE
688 C If Er is constant and the IDD's have been saved use them
689 C DO 92 IL-1,16
690 C IDD(IL)=SVIDD(14,IL)
691 C 92 CONTINUE
692 C ENDP
693 C
694 C
695 C THIS SECTION UTILIZED THE IDD VALUES TO COMPUTE THE MUELLER MTX
696 C ELEMENTS DIVIDED BY Q

```

## Appendix VI

```

697   C P(1)-P11/Q, P(2)-P12/Q .. P(5)-P21/Q .. P(16)-P44/Q
698   C MANY OF THESE P VALUES ARE ZERO BUT THEY ARE CALCULATED HERE FOR
699   C COMPLETENESS. THEIR CALCULATION TIME IS MINUTE COMPARED TO THE ACTUAL
700   C INTEGRATIONS.
701   P(1)=-.5*(IDD(1)+IDD(2)+IDD(3)+IDD(4))
702   P(2)=-.5*(IDD(1)+IDD(2)-IDD(3)-IDD(4))
703   P(3)=-IDD(9)
704   P(4)=-IDD(10)
705   P(5)=-.5*(IDD(1)-IDD(2)+IDD(3)-IDD(4))
706   P(6)=-.5*(IDD(1)-IDD(2)-IDD(3)+IDD(4))
707   P(7)=-IDD(13)
708   P(8)=-IDD(14)
709   P(9)=-2.*IDD(11)
710   P(10)=-2.*IDD(15)
711   P(11)=-IDD(5)+IDD(7)
712   P(12)=-IDD(6)+IDD(8)
713   P(13)=-2.*IDD(12)
714   P(14)=-2.*IDD(16)
715   P(15)=-IDD(6)*IDD(8)
716   P(16)=-IDD(5)-IDD(7)
717   C
718   C The Muller matrix elements are computed in this section. Only the 6
719   C independent matrix elements are passed to the main program.
720
721   C write(8,2001)Q(RCODE)
722   C write(8,2001)P(1),P(2),P(6),P(11),P(12),P(16)
723   IF(RCODE.EQ.0)THEN
724     PP(1)=1.
725     PP(2)=P(2)/P(1)
726     PP(3)=P(6)/P(1)
727     PP(4)=P(11)/P(1)
728     PP(5)=P(12)/P(1)
729     PP(6)=P(16)/P(1)
730   ELSE
731     PP(1)=P(1)*Q(RCODE)
732     PP(2)=P(2)*Q(RCODE)
733     PP(3)=P(6)*Q(RCODE)
734     PP(4)=P(11)*Q(RCODE)
735     PP(5)=P(12)*Q(RCODE)
736     PP(6)=P(16)*Q(RCODE)
737   ENDIF
738   2001 FORMAT(6E12.4)
739   C
740   RETURN
741   END
742   C
743   C
744   C
745   C THIS SUBROUTINE DRIVES QDAG TO COMPUTE Q
746   SUBROUTINE QCMP(Q, AERR, RERR, RCODE)
747   REAL VX, VY, VXZ, VYSQH, UPLIM, QPRIME, SETARG, ARG, ARG1,
748   1 UPLIMD, UPLIM1, SETUPR
749   REAL HMSQ, CLEN, CLENSQ, WLEN, THETA, SNTHT, CSTHT, SIGS, K0, PI
750   REAL GCNTI

```

# Appendix VI

```

751     INTEGER RCODE
752     EXTERNAL ARG
753     COMMON/ONE/HMSQ,CLEN,CLENSQ,WLEN,THETA,SNHTT,CSTHT,SIGS,K0,PI
754     C FOR EACH VALUE OF THETA COMPUTE COMPONENTS OF VECTOR V.
755     C THEN COMBINE WITH THE MEAN SQUARE HEIGHT AND SAVE IN ARG
756     C  WRITE(8,'Y' SNHTT,CSTHT,SNHTT,CSTHT
757     VX=-2.*K0*SNHTT
758     C  VZ=0.
759     VY=2.*K0*CSTHT
760     VXZ=ABS(VX)
761     VYSQH=VY*VY*HMSQ
762     SETARG=ARG1(VXZ,VYSQH)
763     UPLIMD=SETUPR(RCODE)
764     SETARG=GCNT1(0.)
765     C COMPUTE THE UPPERLIMIT ON THE INTEGRATION BY ASSUMING THE INTEGRAND
766     C DIES AWAY WITHIN UPLIMD CORRELATION LENGTHS.
767     UPLIM=UPLIMD*CLEN
768     UPLIM1=UPLIM/100.
769     100 CONTINUE
770     C COMPUTE THE DOUBLE INTEGRAL WHERE ONE HALF -> 2PI
771     C THIS CAN BE DONE BY SWITCHING TO POLAR COORDINATES
772     C IF NO INTERGRAND IS FOUND THEN REDUCE INTEGRATION LIMITS TO FIND IT
773     CALL QDAG(ARG,0.,UPLIM,AERR,RERR,1,QPRIME,ERREST)
774     QPRIME=QPRIME*2*PI
775     IF(QPRIME.NE.1.E-08.AND.UPLIM.GT.UPLIM1) THEN
776     UPLIM=UPLIM*0.7
777     GOTO 100
778     ENDIF
779     C
780     C IF QPRIME > 0. THEN COMPUTE THE TOTAL Q
781     C IF QPRIME STILL EQUALS ZERO THEN INDICATE BY -998.
782     C IF QPRIME WAS LESS THAN ZERO INDICATE BY -999.
783     IF (QPRIME.GT.0.)THEN
784     Q=(K0*K0/PI)*QPRIME
785     ELSE IF(QPRIME.EQ.0.)THEN
786     Q=-998.
787     ELSE IF(QPRIME.LT.0.)THEN
788     Q=-999.
789     ENDIF
790     C  WRITE(8,'Y' Q-'Q' ACCESSED',GCNT1(0.)
791     RETURN
792     END
793     C
794     C
795     C
796     C THIS FUNCTION COMPUTES THE INTEGRAND OF Q FOR DCADRE.
797     C THIS FUNCTION HAS 3 ENTRY POINTS
798     C BSJO COMPUTES BESSEL FUNCTION J SUB 0
799     FUNCTION ARG(RD)
800     REAL BSJO,RSURF,CHT2,CHISQ,JSUB0,RD,VXZ,VYSQH
801     REAL COUNT,DUMMY
802     INTEGER RCODE,DRCODE
803     SAVE
804     COUNT=COUNT+1.

```

# Appendix VI

```

805 JSUBO=BSJO(VXZ*RD)
806 CHZ=EXP((RJSURP(RD)-1.)**VYSQH)
807 ARG=JSUBO*(CHZ-CHISQ)*RD
808 C ARG-1.
809 RETURN
810 C
811 C
812 C THIS SECOND ENTRY POINT SAVES SOME CONSTANTS FOR A GIVEN INTEGRATION
813 C THESE VALUES ARE ONLY FUNCTIONS OF THE SURFACE PARAMETERS NOT OF RD
814 ENTRY ARG1(DVXZ,DVYSQH)
815 VXZ=DVXZ
816 VYSQH=DVYSQH
817 CHISQ=EXP(-VYSQH)
818 ARG1=1.0
819 RETURN
820 C
821 C
822 C THIS ENTRY RETURNS THE NUMBER OF ACCESSES SINCE LAST INQUIRY
823 ENTRY GCNT1(DUMMY)
824 GCNT1=COUNT
825 COUNT=DUMMY
826 RETURN
827 END
828 C
829 C
830 C THIS FUNCTION CALCULATES THE AUTOCORRELATION FUNCTION USED TO
831 C MODEL THE RANDOMLY ROUGH SURFACE
832 C BSK1 COMPUTES THE BESSEL FUNCTION K SUB 1
833 C BSK0 COMPUTES THE BESSEL FUNCTION K SUB 0
834 FUNCTION RJSURP(RD)
835 REAL RD,R,RSQ,R4TH,R6TH,KAPPA6,KAPPA8,BSK1,BSK0,TERM1,TERM2
836 REAL HMSQ,CLEN,CLENSQ,WLEN,THETA,SNHTT,CSTHT,SICS,K0,PI
837 INTEGER RCODE,DRCODE
838 COMMON/ONE/HMSQ,CLEN,CLENSQ,WLEN,THETA,SNHTT,CSTHT,SICS,K0,PI
839 IF(RD.EQ.0.) THEN
840 RJSURP=1.
841 RETURN
842 ENDP
843 IF(RCODE.EQ.1) THEN
844 RJSURP=EXP(-RD**RD/CLENSQ)
845 RETURN
846 ELSE IF(RCODE.EQ.2) THEN
847 R=RD**KAPPA8
848 RSQ=R**R
849 R4TH=RSQ**R4TH
850 R6TH=RSQ**R6TH
851 TERM1=(1.-3.*RSQ/8.+3.*R4TH/32.+R6TH/3072.)*R**BSK1(R)
852 TERM2=(RSQ/2.-R4TH/4.-R6TH/96.)*R**BSK0(R)
853 RJSURP=TERM1+TERM2
854 RETURN
855 ELSE IF(RCODE.EQ.3) THEN
856 R=RD**KAPPA6
857 RSQ=R**R
858 R4TH=RSQ**RSQ

```

# Appendix VI

```

859      TERM1 = (1. - 3. * RSQ/4. - R4TH/96.) * R * BSK1(R)
860      TERM2 = (RSQ/2. + 3. * R4TH/16.) * BSK0(R)
861      RRSURF = TERM1 + TERM2
862      RETURN
863      ELSE
864      WRITE(8,*) 'ILLEGAL RCODE'
865      STOP
866      ENDP
867      RRSURF = 0.
868      RETURN
869      C
870      C
871      ENTRY SETUPR(DRCODE)
872      RCODE = DRCODE
873      SETUPR = 1.
874      IF(RCODE.EQ.1) THEN
875      SETUPR = 5.
876      ELSE IF(RCODE.EQ.2) THEN
877      KAPPA8 = SQRT(1.6) * CLEN
878      SETUPR = 175. / SQRT(1.6)
879      ELSE IF(RCODE.EQ.3) THEN
880      KAPPA6 = 1. * CLEN
881      SETUPR = 175.
882      ENDP
883      RETURN
884      END
885      C
886      C
887      C
888      C THIS SUBROUTINE DRIVES TWODQ TO COMPUTE IDD FOR DIJ * DKL IN INTEGRAND
889      C THE REAL PART IS COMPUTED FOR CODE = 1 OR 2, IMAGINARY CODE = 3
890      SUBROUTINE IDDCMP(IDD, AERR, RERR, IL, EREST)
891      REAL IDD, AERR, RERR, EREST
892      REAL HZMIN, HZMAX, HXMIN, HXMAX, HZMAX1
893      REAL SDPQ, SP2, SPHXHZ, SZMIN, SZMAX
894      REAL HMSQ, CLEN, CLENSQ, WLEN, THETA, SNTHT, CSTHT, SIGS, K0, PI
895      REAL GCVT2
896      INTEGER ICODE(16), ISUB(16), JSUB(16), KSUB(16), LSUB(16)
897      INTEGER IL
898      COMPLEX EREST, UR
899      EXTERNAL ARGIDD, ZMIN, ZMAX
900      COMMON/ONE/HMSQ, CLEN, CLENSQ, WLEN, THETA, SNTHT, CSTHT, SIGS, K0, PI
901      DATA ICODE/1,1,1,1,2,3,2,3,2,3,2,3,2,3,2,3/
902      DATA ISUB/1,1,2,2,1,1,1,1,1,1,1,1,2,2,1,1/
903      DATA JSUB/1,2,1,2,1,1,2,2,1,1,1,1,1,1,2,2/
904      DATA KSUB/1,1,2,2,2,2,2,2,1,1,2,2,2,2,2,2/
905      DATA LSUB/1,2,1,2,2,2,1,1,2,2,1,1,2,2,2,2/
906      C
907      UR = (1., 0.)
908      SETUP = SDPQ(ICODE(IL), ISUB(IL), JSUB(IL), KSUB(IL),
909      1 LSUB(IL), EREST, UR, CSTHT, SNTHT)
910      SETUP = SP2 * SIGS, SNTHT, CSTHT, PI)
911      C WRITE(8,*) 'FACT = ', SETUP
912      SETUP = SPHXHZ(PI, SIGS)

```

# Appendix VI

```

913 C WRITE(8,'YDENOM- ',SETUP
914 SETUP=GCNT2(0.)
915 HZMIN=0.
916 HZMAX=5.*SQRT(SIGS)
917 HZMAX1=HZMAX*.4
918 HXMIN=HZMAX
919 HXMAX=HZMAX
920 199 SETUP=SZMIN(HZMIN)
921 SETUP=SZMAX(HZMAX)
922 C
923 CALL TWODQ(ARGIDD,HXMIN,HXMAX,ZMIN,ZMAX,AERR,RERR,1,IDD,ERREST)
924 IDD=2.*IDD
925 C
926 C WRITE(8,'YIDD- ',IDD,' ACCESSED',GCNT2(0.)
927 IF (IDD.EQ.0.0.AND.HZMAX.GT.HZMAX1)THEN
928 HZMIN=HZMIN*.7
929 HZMAX=HZMAX*.7
930 HXMIN=HXMIN*.7
931 HXMAX=HXMAX*.7
932 GOTO 199
933 ENDIF
934 C
935 RETURN
936 END
937 C
938 C
939 C
940 C THESE FUNCTIONS CALCULATE THE HZ LIMITS FOR 2-D INTEGRATION BY TWODQ
941 FUNCTION ZMIN(X1)
942 REAL D1ZMIN,D2ZMIN
943 SAVE
944 ZMIN=D1ZMIN
945 RETURN
946 ENTRY SZMIN(D2ZMIN)
947 D1ZMIN=D2ZMIN
948 SZMIN=1.0
949 RETURN
950 END
951 C
952 FUNCTION ZMAX(X1)
953 REAL D1ZMAX,D2ZMAX
954 SAVE
955 ZMAX=D1ZMAX
956 RETURN
957 ENTRY SZMAX(D2ZMAX)
958 D1ZMAX=D2ZMAX
959 SZMAX=1.0
960 RETURN
961 END
962 C
963 C
964 C
965 C THIS FUNCTION CALCULATES THE ARGUMENT OF IDD FOR TWODQ
966 FUNCTION ARGIDD(HX,HZ)

```

# Appendix VI

```

967     REAL COUNT,DUMMY,HXZSQ
968     REAL DENOM,DSIGS,SIGS,SNTHT,CSTHT,P1,FACT
969     REAL SIGSRT,PIC,C0,TANT,COTT,R,P1,P2
970     REAL ERPC
971     SAVE
972     C
973     COUNT=COUNT+1.
974     HXZSQ=HX*HX+HZ*HZ
975     C
976     IF (HX.LT.-COTT) THEN
977         P2=0.
978     ELSE
979         P2=FACT
980    ENDIF
981     C
982     PHXHZ=EXP(-HXZSQ/SIGS)/DENOM
983     C
984     ARGIDD=PHXHZ*P2*DPQMAG(HX,HZ,HXZSQ)
985     RETURN
986     C
987     C
988     ENTRY SPHXHZ(P1,DSIGS)
989     SIGS=DSIGS
990     DENOM=P1*SIGS
991     SPHXHZ=DENOM
992     RETURN
993     C
994     C
995     ENTRY SP2(DSIGS,SNTHT,CSTHT,P1)
996     SIGSRT=SQRT(DSIGS)
997     PIC=.5*SIGSRT/SQRT(P1)
998     IF(ABS(SNTHT).LT.1.E-10)THEN
999         COTT=1.E30
1000     C0=0.
1001     ELSE
1002         TANT=SNTHT/CSTHT
1003         COTT=1/TANT
1004         R=COTT/SIGSRT
1005         P1=EXP(-R*R)
1006         P2=ERPC(R)
1007         C0=PIC*TANT*P1-.5*P2
1008     ENDIF
1009     FACT=1./(1.+C0)
1010     SP2=FACT
1011     RETURN
1012     C
1013     C
1014     ENTRY GCNT2(DUMMY)
1015     GCNT2=COUNT
1016     COUNT=DUMMY
1017     RETURN
1018     END
1019     C
1020     C

```

# Appendix VI

```

1021  C
1022  C THIS FUNCTION CALCULATE DQ*DKL AS PART OF IDD'S INTEGRAND
1023  FUNCTION DPQMAG(HX,HZ,HXZSQ)
1024  COMPLEX ER,UR,RIR,ETAR,ERM,ERM,ERM,ERM,URM,URM,URM,URM
1025  COMPLEX SN1N,CS1N,DEN2,DEN3,C1IP,CPVV,CPHH,B1,B2,B3,B4
1026  COMPLEX DPQ(2,2),DER,DUR
1027  REAL HX,HZ,HXZSQ,SNHT,CSHT,DSNHT,DCSTHT
1028  REAL CSG,TNG,SNG,CS0N,SN0N,SNPD1,CSPD1,S0IP,C0IP
1029  REAL CSSI,SNSI,SNSIP
1030  INTEGER ICODE,ICODE,ISUB,ISUB,JSUB,JSUB,KSUB,KSUB,LSUB,LSUB
1031  SAVE
1032  C
1033  CSG=1./SQRT(1.+HXZSQ)
1034  TNG=SQRT(HXZSQ)
1035  SNG=CSG*TNG
1036  IF(TNG.LT.1.E-5) THEN
1037    CS0N=CSHT
1038    SN0N=SNHT
1039  ELSE
1040    CSPD1=-HX/TNG
1041    SNPD1=-HZ/TNG
1042    CS0N=CSG*CSHT-SNG*SNHT*CSPD1
1043    SN0N=SQRT(1.-CS0N*CS0N)
1044  ENDIF
1045  C
1046  IF(ABS(CS0N).LT.1.E-5) THEN
1047    DPQMAG=0.
1048    RETURN
1049  ENDIF
1050  C
1051  SN1N=SN0N/RIR
1052  CS1N=CSQRT(1.-SN1N*SN1N)
1053  DEN2=CS0N+CS1N*ETAR
1054  DEN3=CS0N+CS1N*ETAR
1055  S0IP=SN0N*SN0N
1056  C0IP=CS0N
1057  C1IP=CS1N*CS1N
1058  CPVV=C0IP*((-UR*C1IP-S0IP)*ERM-URM)/(DEN2*DEN2)
1059  CPHH=C0IP*((-ER*C1IP-S0IP)*URM-ERM)/(DEN3*DEN3)
1060  C
1061  IF(TNG.LT.1.E-5) THEN
1062    DPQ(1,1)=CPVV
1063    DPQ(1,2)=0.
1064    DPQ(2,1)=0.
1065    DPQ(2,2)=CPHH
1066  ELSE
1067    CSSI=(CSG*SNHT+SNG*CSHT*CSPD1)/SN0N
1068    SNSI=-SNG*SNPD1/SN0N
1069    SNSIP=-SNSI
1070    B1=CPVV*CSSI
1071    B2=-CPHH*SNSI
1072    B3=CPVV*SNSI
1073    B4=-CPHH*CSSI
1074    DPQ(1,1)=CSSI*B1-SNSIP*B2

```

# Appendix VI

```

1075      DPQ(1,2) = CSSP*B3-SNSIP*B4
1076      DPQ(2,1) = SNSIP*B1+CSSP*B2
1077      DPQ(2,2) = SNSIP*B3+CSSP*B4
1078      ENDP
1079      C
1080      IF(ICODE.EQ.1) THEN
1081        DPQMAG = (CABS(DPQ(ISUB,JSUB))/CSG)**2
1082      ELSE IF(ICODE.EQ.2) THEN
1083        DPQMAG = REAL(DPQ(ISUB,JSUB)*CONJG(DPQ(KSUB,LSUB)))/CSG/CSG
1084      ELSE
1085        DPQMAG = ABMAG(DPQ(ISUB,JSUB)*CONJG(DPQ(KSUB,LSUB)))/CSG/CSG
1086      ENDP
1087      RETURN
1088      C
1089      C
1090      ENTRY SDPQ(ICODE,ISUB,JSUB,KSUB,LSUB,DER,DUR,
1091      1  DCSTHT,DSNTHT)
1092      ICODE = ICODE
1093      ISUB = ISUB
1094      JSUB = JSUB
1095      KSUB = KSUB
1096      LSUB = LSUB
1097      CSTHT = DCSTHT
1098      SNTHT = DSNTHT
1099      ER = DER
1100      UR = DUR
1101      RUR = CSQRT(ER*UR)
1102      ETAR = CSQRT(UR/ER)
1103      ERM = 1./ER
1104      ERMUR = 1./UR
1105      ERMUR = ERMUR*RUR
1106      URM = 1./UR
1107      URMUR = 1./UR
1108      URMUR = URMUR*RUR
1109      SDPQ = 1.0
1110      RETURN
1111      END
1112      C*****END OF RETRO*****
1113      C
1114      C
1115      C *****SUBROUTINE TZSTAT*****
1116      C This subroutine computes the basic statistics used in the analyses.
1117      C For more information see the user manual Equations (1)-(5)
1118      C
1119      SUBROUTINE TZSTAT(DIFF,RATLOG,LMAX,DIFFMAX,RADMAX,
1120      *URMAX,RADIUS,RLOG,PFT,PPB)
1121      DIMENSION DIFF(6),RATLOG(6),PFT(6),PPB(6)
1122      RAD=0.
1123      RADMAX=0.
1124      RADRAT=0.
1125      DIFFMAX=0.
1126      DO 330 L=1,6
1127        IF(ABS(PFT(L)).GT.ABS(PPB(L)))THEN
1128          DIFF(L)=PFT(L)-PPB(L)

```

# Appendix VI

```

1129      IF(FFT(L).NE.0..AND.FFB(L).NE.0.)THEN
1130      RATLOG(L)=DIFF(L)/SQRT(ABS(FFT(L)*FFT(L)))
1131      ELSE
1132      RATLOG(L)=0.
1133      ENDDIF
1134      ELSE
1135      DIFF(L)=0.
1136      RATLOG(L)=0.
1137      ENDDIF
1138      RAD=RAD+DIFF(L)*DIFF(L)
1139      RADRAT=RADRAT+RATLOG(L)*RATLOG(L)
1140      IF(ABS(RATLOG(L)).GT.ABS(RADMAX))THEN
1141      RADMAX=RATLOG(L)
1142      LRMAX=L
1143      ENDDIF
1144      IF(ABS(DIFF(L)).GT.ABS(DIFFMAX))THEN
1145      DIFFMAX=DIFF(L)
1146      LMAX=L
1147      ENDDIF
1148      300 CONTINUE
1149      RAD=SQRT(RAD)
1150      RLOG=SQRT(RADRAT)
1151      C WRITE(8,8800)RADMAX
1152      8800 FORMAT('RADMAX',E12.4)
1153      RETURN
1154      END
1155      C
1156      C*****SUBROUTINE SORTNK*****
1157      C
1158      C The routine sorts through the
1159      C array containing the imaginary part of the index and finds all maxima and
1160      C minima. If there is an interval over which there are local extrema and
1161      C several points share the extreme value, then the program will store the
1162      C extreme value at the two endpoints of the interval, and it is implicitly
1163      C understood that it is constant over that interval. The array containing
1164      C the resonant values (anexima), ANKMAX is rearranged in the last section
1165      C of this routine so that its entries are in order of decreasing k rather
1166      C than in order of increasing wavelength.
1167      C
1168      SUBROUTINE SORTNK(NMIN,NMAX,WLNMN,WLNMIN,ANKMIN,ANKMAX,NPTS,
1169      *NK,WLN)
1170      COMMON/TWO/WLNMN,WLMAX
1171      REAL WLNMN(40),WLNMN(40),ANKMIN(40),ANKMAX(40),WLN(300),NK(300)
1172      INTEGER NMIN,NMAX
1173      C To avoid looking at wavelengths outside of the range defined by WLMAX
1174      C and WLMIN, find out which wavelengths in WLN are outside of the range
1175      C and exclude their corresponding entries in NK.
1176      C PRINT *,WLMIN,WLMAX
1177      WLNMX=0.
1178      ANKMAX=0.
1179      NLESS=1
1180      NMORE=1
1181      DO 777 JL=1,NPTS

```

# Appendix VI

```

1182 IF(WLN(J).GT.WLMAX)NMORE-NMORE+1
1183 777 CONTINUE
1184 C WRITE(8,8222)NLESS,NMORE
1185 8222 FORMAT('NLESS, NMORE',2I5)
1186 NMIN=0
1187 NMAX=0
1188 IF(NK(NLESS).GT.NK(NLESS+1))THEN
1189 NMAX=NMAX+1
1190 WLNMX(NMAX)-WLN(NLESS)
1191 ANKMAX(NMAX)-NK(NLESS)
1192 ELSE
1193 NMIN=NMIN+1
1194 WLNMIN(NMIN)-WLN(NLESS)
1195 ANKMIN(NMIN)-NK(NLESS)
1196 ENDIF
1197 M=NLESS+1
1198 C WRITE(8,*)NMORE,NLESS
1199 99 DO 301 J=M,NPTS-NMORE
1200 IF(NK(J).GT.NK(J-1).AND.NK(J).GT.NK(J+1))THEN
1201 NMAX=NMAX+1
1202 WLNMX(NMAX)-WLN(J)
1203 ANKMAX(NMAX)-NK(J)
1204 ELSE IF(NK(J).LT.NK(J-1).AND.NK(J).LT.NK(J+1))THEN
1205 NMIN=NMIN+1
1206 WLNMIN(NMIN)-WLN(J)
1207 ANKMIN(NMIN)-NK(J)
1208 C
1209 C Careful about drawing a false conclusion if two neighboring entries in
1210 C the data file are equal.
1211 C
1212 ELSE IF(NK(J).GT.NK(J-1).AND.NK(J).EQ.NK(J+1))THEN
1213 NMAX=NMAX+1
1214 WLNMX(NMAX)-WLN(J)
1215 ANKMAX(NMAX)-NK(J)
1216 JJ=J
1217 MM=1
1218 DO 302 MMM=J+1,NPTS-1
1219 MM=MM+1
1220 IF(NK(MMM).GT.NK(MMM+1))THEN
1221 NMAX=NMAX+1
1222 WLNMX(NMAX)-WLN(MMM)
1223 ANKMAX(NMAX)-NK(MMM)
1224 GO TO 303
1225 ENDIF
1226 IF(NK(MMM).LT.NK(MMM+1))NMAX=NMAX-1
1227 IF(NK(MMM).LT.NK(MMM+1))GO TO 303
1228 302 CONTINUE
1229 ELSE IF(NK(J).LT.NK(J-1).AND.NK(J).EQ.NK(J+1))THEN
1230 NMIN=NMIN+1
1231 WLNMIN(NMIN)-WLN(J)
1232 ANKMIN(NMIN)-NK(J)
1233 JJ=J
1234 MM=1
1235 DO 305 MMM=J+1,NPTS-1

```

# Appendix VI

```

1236      MM-MM+1
1237      IF(NK(MMM).LT.NK(MMM+1))THEN
1238      NMIN-NMIN+1
1239      WLNMIN(NMIN)-WLN(MMM)
1240      ANKMIN(NMIN)-NK(MMM)
1241      GO TO 303
1242      ENDIF
1243      IF(NK(MMM).GT.NK(MMM+1))NMIN-NMIN-1
1244      IF(NK(MMM).GT.NK(MMM+1))GO TO 303
1245 305 CONTINUE
1246      ENDIF
1247 301 CONTINUE
1248      IF(NK(NPTS-NMORE+1).GT.NK(NPTS-NMORE))THEN
1249      NMAX-NMAX+1
1250      WLNMX(NMAX)-WLN(NPTS-NMORE+1)
1251      ANKMAX(NMAX)-NK(NPTS-NMORE+1)
1252      ELSE
1253      NMIN-NMIN+1
1254      WLNMIN(NMIN)-WLN(NPTS-NMORE+1)
1255      ANKMIN(NMIN)-NK(NPTS-NMORE+1)
1256      ENDIF
1257      GO TO 304
1258 303 M-JJ+MM
1259      GO TO 99
1260      C
1261      C Rearrange the elements in WLNMAX so that they are in decreasing order
1262      C rather than in order of increasing wavelength. If a case occurs where
1263      C two elements of WLNMAX are equal, handle it by setting the one with the
1264      C largest subscript equal to zero.
1265      C
1266 304 DO 321 I=1,NMAX-1
1267      LAST=NMAX-I
1268      DO 321 K=1,LAST
1269      IF(ANKMAX(K).EQ.ANKMAX(K+1))ANKMAX(K+1)=0.
1270      IF(ANKMAX(K).LT.ANKMAX(K+1))THEN
1271      TEMP=ANKMAX(K)
1272      TEMPWL=WLNMX(K)
1273      ANKMAX(K)=ANKMAX(K+1)
1274      WLNMX(K)=WLNMX(K+1)
1275      ANKMAX(K+1)=TEMP
1276      WLNMX(K+1)=TEMPWL
1277      ENDIF
1278 321 CONTINUE
1279      RETURN
1280      END
1281      C
1282      C====SUBROUTINE SAVE====
1283      C This routine determines whether or not the data computed by RETRO is more
1284      C useful from the standpoint of the test specified by NCRIT than the currently
1285      C most useful set of data. Since discrepancies arise between the 2 tests,
1286      C performed by TZSTAT, the variable NCRIT determines which test is used in
1287      C decision making.
1288      C
1289      SUBROUTINE SAVE(NCRIT,TZSTMX,RADMX,D,M,DMX,WMX,DRDMX,WRDMX

```

# Appendix VI

```

1290      *,JWLSET,RADIUS,RLOG)
1291 C  WRITE(8,888)I3,I4,RADIUS,RLOG
1292 888 FORMAT('FROM SAVE',2I5,2E12.4)
1293      JWLSET-1
1294      IF(RADIUS.GE.TZSTMX)THEN
1295          TZSTMX-RADIUS
1296          I3MX-I3
1297          I4MX-I4
1298          IF(NCRIT.EQ.1)JWLSET-2
1299      ENDIF
1300      IF(RLOG.GE.RADMX)THEN
1301          RADMX-RLOG
1302          I3RDMX-I3
1303          I4RDMX-I4
1304          IF(NCRIT.EQ.0)JWLSET-2
1305      ENDIF
1306      RETURN
1307      END
1308 C
1309 C*****SUBROUTINE WLANGSET*****
1310 C
1311 C This routine checks if Muller matrices have been computed for the point
1312 C at wavelength W and incident angle ANG(K+I4SET). If Muller matrices have
1313 C already been calculated for this point, then the program will not recompute
1314 C them (NCALC-2). If the program has yet to compute Muller matrices for this
1315 C wavelength and incident angle, it will (NCALC-1).
1316 C
1317 SUBROUTINE WLANGSET(W,MWLN,WLNWLN,WLN,I,K,I3SET,I4SET,NCALC,
1318      *I3,I4)
1319 REAL WLNWLN(100),WLN(100,100)
1320 NCALC-2
1321 KKK-I4SET+K
1322 C If J=0 then some incident angles have already been computed for
1323 C this wavelength. It is hence necessary to determine if ANG(K+I4SET)
1324 C has been used at this wavelength. The wavelengths and incident angles
1325 C at which Muller matrices for those wavelengths have been calculated
1326 C are stored in WLN so if WLN(...) is zero then that Muller matrix has
1327 C not yet been computed.
1328 IF(J.EQ.0)THEN
1329     IF(WLN(I3SET,KKK).LT.1.E-05)THEN
1330         WLN(I3SET,KKK)-W
1331         I3-I3SET
1332         I4-KKK
1333         NCALC-1
1334         RETURN
1335     ELSE
1336         RETURN
1337     ENDIF
1338 ELSE
1339 C If J.NE.0 then there is a chance that this wavelength has not been used
1340 C at all in which case there is no chance that the same Muller matrix will
1341 C be computed twice and any incident angle is o.k.
1342 DO 99 I=1,MWLN
1343     X=W-WLNWLN(I)

```

# Appendix VI

```

1344     IF(ABS(X).LT.1.E-05)THEN
1345     IF(WLN(L,KKK).LT.1.E-05)THEN
1346     WLN(L,KKK)-W
1347     D-L
1348     W-KKK
1349     NCALC-1
1350     RETURN
1351     ELSE
1352     RETURN
1353     ENDP
1354     ENDP
1355     99 CONTINUE
1356     ENDP
1357     C If this line has been reached then the wavelength W has yet to be
1358     C used.
1359     MWLN-MWLN+1
1360     WLNWLN(MWLN)-W
1361     WLN(MWLN,KKK)-W
1362     D-MWLN
1363     W-KKK
1364     NCALC-1
1365     RETURN
1366     END
1367     C
1368     C*****SUBROUTINE MTXLE*****
1369     C
1370     C This routine sorts the arrays containing the terms in the discrimination
1371     C tests of T2STAT in order of largest to smallest. The purpose of this is
1372     C to tell you which matrix elements are most useful at a particular
1373     C angle of incidence and wavelength.
1374     C
1375     SUBROUTINE AMTXLE(X,Y,K,N)
1376     INTEGER K(6),KK(6)
1377     REAL X(6),Y(6)
1378     DATA KK/1,2,3,4,5,6/
1379     Y-X
1380     K-KK
1381     C WRITE(8,8888)Y
1382     C WRITE(8,8888)X
1383     8888 FORMAT(6E12.4)
1384     DO 100 M-1,5
1385     DO 100 L-1,6-M
1386     IF(ABS(Y(L)).LE.ABS(Y(L+1)))THEN
1387     T=Y(L)
1388     Y(L)-Y(L+1)
1389     Y(L+1)-T
1390     JT=K(L)
1391     K(L)-K(L+1)
1392     K(L+1)-JT
1393     ENDP
1394     100 CONTINUE
1395     N-0
1396     DO 101 J-1,6
1397     IF(Y(J).NE.0)N=N+1

```

# Appendix VI

```

1398 101 CONTINUE
1399 RETURN
1400 END
1401 C
1402 C*****SUBROUTINE SETHETWL*****
1403 C
1404 C This routine determines which angles of incidence and wavelengths
1405 C the program uses to compute and analyze Mueller matrices in an
1406 C orderly fashion so they can be written in a form that
1407 C DISPLAY can read and plot them.
1408 C
1409 SUBROUTINE SETHETWL(ANG,WLNWLN,ISET,I4SET)
1410 DIMENSION ANG(100),WLNWLN(100)
1411 COMMON/TWO/WLMIN,WLMAX
1412 A=ANG(I4SET)
1413 IF(A.LT.20.)THEN
1414 DO 666 I=1,11
1415 ANG(I)=4.*(I-1)
1416 666 CONTINUE
1417 ELSE IF(A.GT.60.)THEN
1418 DO 667 I=1,11
1419 ANG(I)=48.+4.*(I-1)
1420 667 CONTINUE
1421 ELSE
1422 DO 671 I=-5,5
1423 ANG(I)=A+4.*I
1424 671 CONTINUE
1425 ENDP
1426 W=WLNWLN(ISET)
1427 IF(W+.25.LE.WLMIN)THEN
1428 X=W-WLMIN
1429 M=IPDX(X/.05)
1430 DO 668 I=-M,-M+10
1431 WLNWLN(I+M+1)=W+P*.05
1432 668 CONTINUE
1433 ELSE IF((W+.25)GT.WLMAX)THEN
1434 X=W-WLMAX-W
1435 M=IPDX(X/.05)
1436 DO 669 I=-M,-M+10
1437 WLNWLN(I+M+1)=W-.05*I
1438 669 CONTINUE
1439 ELSE
1440 DO 670 I=-5,5
1441 WLNWLN(I+6)=W+.05*I
1442 670 CONTINUE
1443 ENDP
1444 RETURN
1445 END

```

## Appendix VI

### AVI.2.6 SAMPLE CALCULATIONS: INPUT AND OUTPUT DATA OUTPUT FILES.

#### AVI.2.6.1 Input File "DATAIN2."

```
1 compos.nk
2 sf96.nk
3 decide2.out
4 decide2.comm2
5 compos
6 sf96
7 20. .5
8 20. .5
9 11 0. 8.
10 9.0 12.5 .05
11 1
12 1
13 .00000001 .0000001
14 .0001 .005
15 700
16 0
17 dic.d
18 cdi.d
19 acdi.d
```

#### AVI.2.6.2 Output File "dic.d."

```
1 sf96
2 0001
3 0001
4 0011
5 0011
6 0.2000E+02
7 0.5000E+00
8 0.1195E+02 0.1200E+02 0.1205E+02 0.1210E+02 0.1215E+02
9 0.1220E+02 0.1225E+02 0.1230E+02 0.1235E+02 0.1240E+02
10 0.1245E+02
11 0.0000E+00 0.4000E+01 0.8000E+01 0.1200E+02 0.1600E+02
12 0.2000E+02 0.2400E+02 0.2800E+02 0.3200E+02 0.3600E+02
13 0.4000E+02
14 0.1000E+01 0.1000E+01 0.1000E+01
15 0.5562E-01 -0.2157E-13 0.5521E-01 -0.5480E-01 0.1661E-11 -0.5521E-01
16 0.1000E+01 0.1000E+01 0.1000E+01
17 0.5552E-01 0.8135E-04 0.5510E-01 -0.5467E-01 -0.6637E-04 -0.5510E-01
18 0.1000E+01 0.1000E+01 0.1000E+01
19 0.5519E-01 0.3183E-03 0.5473E-01 -0.5427E-01 -0.2597E-03 -0.5473E-01
20 0.1000E+01 0.1000E+01 0.1000E+01
21 0.5451E-01 0.6884E-03 0.5399E-01 -0.5348E-01 -0.5617E-03 -0.5399E-01
22 0.1000E+01 0.1000E+01 0.1000E+01
.
.
.
.
```

## Appendix VI

```

.
.
.
.
249 0.5398E+00 0.6587E-01 0.4800E+00 -0.4222E+00 -0.3128E-01 -0.4820E+00
250 0.1000E+01 0.1000E+01 0.1000E+01
251 0.4630E+00 0.6710E-01 0.4059E+00 -0.3522E+00 -0.3167E-01 -0.4092E+00
252 0.1000E+01 0.1000E+01 0.1000E+01
253 0.3753E+00 0.6222E-01 0.3245E+00 -0.2781E+00 -0.2926E-01 -0.3289E+00
254 0.1000E+01 0.1000E+01 0.1000E+01
255 0.2836E+00 0.5238E-01 0.2418E+00 -0.2051E+00 -0.2460E-01 -0.2468E+00

```

### AVI.2.6.3 Output File "cdi.d."

```

1 compos
2 0001
3 0001
4 0011
5 0011
6 0.2000E+02
7 0.5000E+00
8 0.1195E+02 0.1200E+02 0.1205E+02 0.1210E+02 0.1215E+02
9 0.1220E+02 0.1225E+02 0.1230E+02 0.1235E+02 0.1240E+02
10 0.1245E+02
11 0.0000E+00 0.4000E+01 0.8000E+01 0.1200E+02 0.1600E+02
12 0.2000E+02 0.2400E+02 0.2800E+02 0.3200E+02 0.3600E+02
13 0.4000E+02
14 0.1000E+01 0.1000E+01 0.1000E+01
15 0.2454E+00 -0.1181E-11 0.2394E+00 -0.2335E+00 -0.2653E-10 -0.2394E+00
16 0.1000E+01 0.1000E+01 0.1000E+01
17 0.2447E+00 0.7180E-03 0.2386E+00 -0.2325E+00 -0.7195E-04 -0.2386E+00
18 0.1000E+01 0.1000E+01 0.1000E+01
19 0.2426E+00 0.2798E-02 0.2360E+00 -0.2295E+00 -0.2800E-03 -0.2360E+00
20 0.1000E+01 0.1000E+01 0.1000E+01
21 0.2385E+00 0.6014E-02 0.2313E+00 -0.2241E+00 -0.6000E-03 -0.2313E+00
22 0.1000E+01 0.1000E+01 0.1000E+01
.
.
.
.
.

```

```

249 0.1381E+00 0.1315E-01 0.1323E+00 -0.1265E+00 -0.3216E-02 -0.1323E+00
250 0.1000E+01 0.1000E+01 0.1000E+01
251 0.1190E+00 0.1346E-01 0.1133E+00 -0.1078E+00 -0.3279E-02 -0.1135E+00
252 0.1000E+01 0.1000E+01 0.1000E+01
253 0.9686E-01 0.1250E-01 0.9171E-01 -0.8690E-01 -0.3039E-02 -0.9204E-01
254 0.1000E+01 0.1000E+01 0.1000E+01
255 0.7345E-01 0.1052E-01 0.6916E-01 -0.6530E-01 -0.2552E-02 -0.6958E-01

```

### AVI.2.6.4 Output File "acd.d."

## Appendix VI

```

1 compos
2 0001
3 0001
4 0011
5 0011
6 0.2000E+02
7 0.5000E+00
8 0.1195E+02 0.1200E+02 0.1205E+02 0.1210E+02 0.1215E+02
9 0.1220E+02 0.1225E+02 0.1230E+02 0.1235E+02 0.1240E+02
10 0.1245E+02
11 0.0000E+00 0.4000E+01 0.8000E+01 0.1200E+02 0.1600E+02
12 0.2000E+02 0.2400E+02 0.2800E+02 0.3200E+02 0.3600E+02
13 0.4000E+02
14 0.1000E+01 0.1000E+01 0.1000E+01
15 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
16 0.1000E+01 0.1000E+01 0.1000E+01
17 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
18 0.1000E+01 0.1000E+01 0.1000E+01
19 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
20 0.1000E+01 0.1000E+01 0.1000E+01
21 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
22 0.1000E+01 0.1000E+01 0.1000E+01
.
.
.
.
.
249 0.7040E+00 0.4320E+01 0.4018E+00 -0.2798E+01 0.4018E+00 0.1472E+01
250 0.1000E+01 0.1000E+01 0.1000E+01
251 0.5957E+00 0.4291E+01 0.3440E+00 -0.2786E+01 0.3440E+00 0.1466E+01
252 0.1000E+01 0.1000E+01 0.1000E+01
253 0.4771E+00 0.4270E+01 0.2785E+00 -0.2781E+01 0.2785E+00 0.1461E+01
254 0.1000E+01 0.1000E+01 0.1000E+01
255 0.3566E+00 0.4256E+01 0.2101E+00 -0.2782E+01 0.2101E+00 0.1456E+01

```

### AVI.2.6.5 Output File "decide2.comm2."

PROGRAM STOPPED BECAUSE NO IMPROVEMENT IN THE  
RESULTS OF ROUTINE T2STAT HAS BEEN DETECTED

### AVI.2.6.6 Output File "decide2.out."

```

1 compos      sf96
2 20.0, 0.5
3 20.0, 0.5
4 0.9000E+01 0.1250E+02 0.5000E-01
5 0.0000E+00 0.8000E+01 0.1600E+02 0.2400E+02 0.3200E+02
6 0.4000E+02 0.4800E+02 0.5600E+02 0.6400E+02 0.7200E+02
7 0.8000E+02
8 CODE FOR AUTO-CORR FUNCTIONS AND NORMALIZATION 1 1

```

# Appendix VI

```

9   NCRIT  0
10  WAVELENGTH= 0.961539E+01 INCIDENT ANGLE= 0.000000E+00
11  5
12  5
13  0.5633E-09 0.1773E+02    0005 -0.5633E-09    0005 -0.1773E+02
.
.
.
.
.
130 WAVELENGTH= 0.124533E+02 INCIDENT ANGLE= 0.640000E+02
131 5 2 1 6 3 4
132 1 6 3 4 2 5
133 0.4224E-02 0.4238E+01    0001 0.2582E-02    0005 -0.2845E+01
134 WAVELENGTH= 0.124533E+02 INCIDENT ANGLE= 0.720000E+02
135 5 2 1 6 3 4
136 1 6 3 4 2 5
137 0.1444E-03 0.4220E+01    0001 0.8904E-04    0005 -0.2852E+01
138 WAVELENGTH= 0.124533E+02 INCIDENT ANGLE= 0.800000E+02
139 5 2 1 6 3 4
140 1 6 3 4 2 5
141 0.6171E-06 0.4182E+01    0001 0.3839E-06    0005 -0.2841E+01
142 WAVELENGTH= 0.956539E+01 INCIDENT ANGLE= 0.000000E+00
143 5
144 5
145 0.1975E-09 0.1138E+02    0005 -0.1975E-09    0005 -0.1138E+02
146 WAVELENGTH= 0.956539E+01 INCIDENT ANGLE= 0.800000E+01
147
148
149 0.0000E+00 0.0000E+00    0005 0.0000E+00    0005 0.0000E+00
150 WAVELENGTH= 0.966539E+01 INCIDENT ANGLE= 0.000000E+00
151 5
152 5
153 0.6568E-09 0.1835E+02    0005 -0.6568E-09    0005 -0.1835E+02
154 WAVELENGTH= 0.966539E+01 INCIDENT ANGLE= 0.800000E+01
155
156
157 0.0000E+00 0.0000E+00    0005 0.0000E+00    0005 0.0000E+00
158 WAVELENGTH= 0.971539E+01 INCIDENT ANGLE= 0.000000E+00
159 5
160 5
161 0.8643E-09 0.1759E+02    0005 -0.8643E-09    0005 -0.1759E+02
162 WAVELENGTH= 0.971539E+01 INCIDENT ANGLE= 0.800000E+01
163
164
165 0.0000E+00 0.0000E+00    0005 0.0000E+00    0005 0.0000E+00
166 THE MOST PROMISING WAVELENGTH AND ANGLE PAIR IS 0.9665E+01 0.0000E+00
167 0.956539E+01 0.000000E+00
168 0.2823E+00 -0.2309E-11 0.2741E+00 -0.2658E+00 -0.1959E-09 -0.2741E+00
169 0.1944E+01 0.1177E-08 0.1743E+01 -0.1543E+01 0.1537E-11 -0.1743E+01
.
.
.

```

# Appendix VI

270 0.1266E+00 0.2852E-01 0.1050E+00 -0.8727E-01 -0.1246E-01 -0.1089E+00  
271 0.3218E-01 0.5420E-02 0.2998E-01 -0.2813E-01 -0.1361E-02 -0.3034E-01  
272 0.124688E+02 0.560000E+02  
273 0.3198E-01 0.8327E-02 0.2593E-01 -0.2120E-01 -0.3660E-02 -0.2725E-01  
274 0.8155E-02 0.1569E-02 0.7530E-02 -0.7033E-02 -0.3944E-03 -0.7659E-02  
275 0.124688E+02 0.640000E+02  
276 0.3662E-02 0.1095E-02 0.2911E-02 -0.2333E-02 -0.4840E-03 -0.3083E-02  
277 0.9366E-03 0.2051E-03 0.8583E-03 -0.7973E-03 -0.5162E-04 -0.8757E-03  
278 0.124688E+02 0.720000E+02  
279 0.1272E-03 0.4365E-04 0.9951E-04 -0.7751E-04 -0.1933E-04 -0.1052E-03  
280 0.3265E-04 0.8156E-05 0.2973E-04 -0.2740E-04 -0.2055E-05 -0.3033E-04  
281 0.124688E+02 0.800000E+02  
282 0.5544E-06 0.2175E-06 0.4282E-06 -0.3204E-06 -0.9612E-07 -0.4466E-06  
283 0.1429E-06 0.4078E-07 0.1294E-06 -0.1180E-06 -0.1027E-07 -0.1315E-06